

# Automatisierte, mehrdimensionale Kompatibilitätstests zum Aufbau von serviceorientierten Informationssystemen

Johannes Maria Zaha\*, Bettina Bazijanec\*\*

\*Business Process Management Research Group  
Queensland University of Technology  
126 Margaret Street  
GPO Box 2432  
Brisbane, Australia  
j.zaha@qut.edu.au

\*\*Lehrstuhl für Wirtschaftsinformatik und Systems Engineering  
Universität Augsburg  
Universitätsstrasse 16  
86135 Augsburg, Germany  
bettina.bazijanec@wiwi.uni-augsburg.de

**Abstract:** Eine wichtige Voraussetzung für den erfolgreichen Aufbau serviceorientierter Informationssysteme liegt in der effizienten Suche und Einbindung von angebotenen Diensten. Hierzu trägt ein automatisierter Vergleich von Anforderungen und der angebotenen Funktionalität potenziell geeigneter Dienste effizienzsteigernd bei. Einem automatisierten Abgleich von Spezifikationen stehen jedoch Hemmnisse entgegen. Zumeist werden Dienste lediglich auf technischer Ebene spezifiziert und fachliche Anforderungen bzw. Einschränkungen, die sich aus der Anwendungsdomäne ergeben, nicht miteinbezogen. Ebenso fehlen geeignete Testmethoden, die einen mehrdimensionalen Abgleich von Spezifikationen über technische und fachliche Ebenen hinweg erlauben. Der vorliegende Beitrag diskutiert die fachliche Kompatibilität von Dienstspezifikationen und leitet ein Kompatibilitätsmaß her, das zur Bewertung von Vergleichsspezifikationen im Rahmen von automatisierten Kompatibilitätstests verwendet wird. Hierzu wird vorgeschlagen, die für fachliche Black-Box-Komponenten verwendeten Spezifikationsmethoden für die Beschreibung von Diensten heranzuziehen, um darauf basierend das Kompatibilitätsmaß zu definieren. Schließlich wird die Vorgehensweise bei mehrdimensionalen Kompatibilitätstests beschrieben.

## 1 Kompatibilität in serviceorientierten Architekturen

Mit den unter dem Begriff *Serviceorientierung* diskutierten Ansätzen finden verteilte Informationssysteme auf Basis entfernter Funktionsaufrufe neue Beachtung. Dies wurde insbesondere durch die Standardisierungserfolge im Bereich der Web Services ermöglicht. Als Web Service wird eine zusammengehörende Menge von vermarktbareren Diensten bezeichnet, die einem autorisierten Nutzerkreis unter Nutzung standardisierter

Kommunikationsprotokolle über wohldefinierte Schnittstellen angeboten werden. Web Services umfassen Schnittstellenbeschreibungen der von ihnen angebotenen Dienste, die in einer standardisierten Beschreibungssprache verfasst sind. Implementierungsdetails bleiben dem Nutzer verborgen. Insofern ähneln sie dem bekannten Konzept der fachlichen Black-Box-Komponenten. Der Unterschied liegt allerdings im Gegenstand der Wiederverwendung und Vermarktung: Bei Web Services ist dies die Leistung, bei Komponenten die Implementierung im Sinne des Leistungserbringers [KrTu02, 2f.]. Im allgemeinen Anwendungsszenario für Web Services stellt ein Anbieter die Beschreibung der von ihm angebotenen Dienste in einem zentralen Verzeichnis bereit und erlaubt so die Suche durch potenzielle Nachfrager. Die Einbindung der gefundenen Dienste in das Zielsystem erfolgt dann mittels standardisierter Funktionsaufrufe über das Internet. Die Beschreibung der Dienste im dargestellten Szenario beschränkt sich allerdings auf die Definition der technischen Schnittstellen, d.h. der benötigten Datentypen und Signaturen vorhandener Dienstoperationen [CCCD01, 19f.]. Sie sind nicht geeignet, fachliche Eigenschaften von Diensten zu spezifizieren [KrTu02, 14]. Dies ist jedoch notwendig, da selbst bei exakt übereinstimmenden, technischen Schnittstellen eine Einbindung von Diensten verhindert wird, wenn der fachliche Inhalt nicht übereinstimmt. Somit ist es auch nicht möglich abzuschätzen, welchen Integrationsaufwand die Kopplung an einen bestimmten Dienst mit sich bringt. *Integration* bezeichnet dabei allgemein die (Wieder-)Herstellung eines Ganzen durch das Zusammenführen verschiedener, jedoch logisch zusammengehörender Teile, ggf. durch Vermittlung [Kaib02, 10; Fran94, 22]. Eine effiziente Vermittlung kann nur gelingen, wenn eine grundsätzliche *Kompatibilität* zwischen den Schnittstellen der einzelnen Systemteile besteht, d.h. wenn zwischen angebotener und nachgefragter Funktionalität ein notwendiges Maß an Übereinstimmung herrscht bzw. wenn vorhandene Unterschiede in der informationstechnischen Darstellung überbrückbar sind. Die Suche nach Diensten lässt sich somit als *mehrdimensionaler Kompatibilitätstest* beschreiben, der über technische und fachliche Beschreibungsebenen hinweg den Grad der Übereinstimmung feststellt und, wenn möglich, eine passende Vermittlungseinheit zur Überbrückung vorhandener, technischer Inkompatibilitäten erzeugt. Die fachliche Übereinstimmung dient als Entscheidungskriterium für den Abbruch des Tests, da fehlende Kompatibilität in der Domäne nicht überbrückt werden kann. Zur Überbrückung technischer Inkompatibilitäten bei der Integration von Black-Box-Komponenten in (dezentralen) Anwendungsarchitekturen existieren allerdings bereits Lösungskonzepte und Methoden wie beispielsweise Adapter, Mediatoren oder Methoden des Datenmappings [YeSt97; Wied95; GaAO95; HoWo04; RaTu01]. Da diese Methoden auf der Abbildung von Schnittstellenelementen, d.h. Operationen und Datentypen, basieren, die sich ebenso in Web Service-Spezifikationen wieder finden, sind sie auch dort anwendbar [KrTu02, 8ff.]. Für die Bestimmung fachlicher Kompatibilität kann, wie bereits erwähnt, nicht auf bestehende Beschreibungsstandards für Web Services zurückgegriffen werden. Hier können jedoch Ansätze aus dem Bereich der komponentenorientierten Softwareentwicklung und dort insbesondere der Spezifikation von Fachkomponenten herangezogen werden. Die dort eingeführte *mehrdimensionale Spezifikation* von Diensten kann dann als Basis für die geforderten Kompatibilitätstests dienen, da sie eine Berücksichtigung sowohl technischer als auch fachlicher Eigenschaften ermöglicht. Da die Bestimmung von Kompatibilität auf technischer Ebene schon gut durchdrungen ist, jedoch noch nicht auf fachlicher Ebene, soll die Bestimmung fachli-

cher Kompatibilität im Folgenden im Vordergrund stehen. Ausgehend von dieser Erkenntnis wird in diesem Beitrag ein *Kompatibilitätsmaß* hergeleitet, das den Übereinstimmungsgrad von zwei fachlichen Spezifikationen angibt und zur Bewertung dieser Dimension im Rahmen mehrdimensionaler Kompatibilitätstests geeignet ist.

In Abschnitt 2 wird zunächst erläutert, welche Aspekte in einem mehrdimensionalen Kompatibilitätstest berücksichtigt werden, wie diese spezifiziert werden können und welche Abhängigkeiten zwischen den Beschreibungsebenen bzw. Dimensionen bestehen. Abschnitt 3 beinhaltet grundsätzliche Überlegungen zur fachlichen Kompatibilität auf Basis der vorgestellten Spezifikation, wie etwa die Zulässigkeit gewisser Namens- und Strukturabweichungen bei der Messung von Kompatibilität. Es werden Übereinstimmungskennzahlen für fachliche Aufgaben und die darin verwendeten Informationsobjekte hergeleitet und zu einem Kompatibilitätsmaß verdichtet. In Abschnitt 4 wird das Vorgehensmodell zur Durchführung automatisierter, mehrdimensionaler Kompatibilitätstests vorgestellt und neben dem fachlichen auch die nachgelagerten technischen Tests erläutert.

## 2 Identifikation der relevanten Beschreibungsdimensionen

Die Beschreibung des Außenverhaltens von Softwareartefakten (Komponenten als auch Services) ist eine komplexe Aufgabe. Es sind sowohl funktionale als auch nicht-funktionale Eigenschaften zu spezifizieren, um so die Prüfung aller für eine Verwendung relevanten Eigenschaften durch einen Nachfrager zu ermöglichen. Um diese Beschreibungskomplexität zu reduzieren, wird bei der Spezifikation von Softwarekomponenten auf verschiedene Beschreibungsebenen zurückgegriffen, die einerseits jeweils nur bestimmte Aspekte spezifizieren aber andererseits zusammen betrachtet eine konsistente Beschreibung der Gesamtkomponente ergeben [Turo02; ABBK02]. Aufgrund noch fehlender Spezifikationsframeworks für Web Services werden im Folgenden etablierte Ansätze aus dem Bereich der Softwarekomponenten verwendet, die sich aber, wie in Abschnitt 1 und den darin zitierten Literaturquellen erläutert, auf Web Services übertragen lassen. In [Turo02] werden für die Spezifikation sieben Ebenen vorgeschlagen, von denen im Folgenden die fünf relevanten Ebenen zur funktionalen Spezifikation erklärt werden (Beispiele zu den Notationen finden sich in Abbildung 2.2):

- Auf *Schnittstellenebene* werden Datentypen vereinbart, Schnittstellensignaturen definiert, sowie Fehlermeldungen und Ausnahmezustände deklariert. Es werden sowohl angebotene als auch benötigte Dienste spezifiziert. Die Spezifikation erfolgt mit Hilfe der Interface Definition Language (IDL) von der Object Management Group (OMG) [OMG02]. Ziel der Vereinbarungen auf Schnittstellenebene ist die Herstellung technischer Kommunikationsfähigkeit zwischen Dienstnehmer und Dienstgeber, wobei syntaktische Aspekte im Vordergrund stehen.
- Die Vereinbarungen auf *Verhaltensebene* dienen der vollständigen Beschreibung des Verhaltens einer Fachkomponente. Hierzu werden Vor- und Nachbedingungen sowie Invarianten mittels Object Constraint Language (OCL) [OMG01] spezifiziert.

- Auf der *Abstimmungsebene* werden Reihenfolgebeziehungen zwischen Dienstoperationen und Synchronisationserfordernisse spezifiziert. Bedingungen auf der Abstimmungsebene können sich sowohl auf die von der Komponente angebotenen als auch die von ihr benötigten Operationen beziehen. Die Verhaltens- und Abstimmungsebene werden mit Hilfe der Object Constraint Language (OCL) [OMG01] bzw. ihrer Erweiterung um temporale Operatoren beschrieben.
- Die *Terminologieebene* dient als zentrales Begriffsverzeichnis und speichert in einem Lexikon alle Begriffe, die für die Spezifikation von Nutzen sind, sowie deren Definition. Es wird somit die Semantik der Fachbegriffe festgelegt.
- Die Dokumentation der von einer Fachkomponente unterstützten betrieblichen Aufgabe sowie ggf. deren Zerlegung in mehrere Teilaufgaben auf fachlichem (konzeptionellem) Niveau wird auf der *Aufgabenebene* vorgenommen. Diese Spezifikation soll es entsprechenden Fachleuten ermöglichen, die Einsatzmöglichkeiten der Komponente bzw. der Dienste in einem konkreten Anwendungsfall zu beurteilen.

Für die fachliche Spezifikation auf Terminologie- und Aufgabenebene wird die Verwendung einer *rekonstruierten Fachnormsprache* vorgeschlagen, die aus einem Lexikon geklärter Fachbegriffe und einer Sammlung von Aussagen besteht, deren Sätze auf vordefinierten Satzbauplänen basieren [Ortn97]. Diese Satzbaupläne ergeben sich aus dem Ziel, Aufgaben einer betrieblichen Anwendungsdomäne zu beschreiben. Da diese Aufgabenspezifikation typischerweise durch Mitarbeiter in Fachabteilungen erfolgt, die über keine tiefer gehenden Kenntnisse in Modellierung und formaler Spezifikation verfügen, wird diese der natürlichen Sprache nahe kommende Spezifikationsmethode gewählt. Eine betriebliche Aufgabe hat das Ziel, einen bestimmten Zustand zu verwirklichen (vgl. [Wöhe96, S. 183]), wobei dazu bestimmte Objekte benötigt und erzeugt werden. Die informationellen Abbilder dieser Objekte werden im Folgenden als Informationsobjekte bezeichnet und können entweder Input- oder Output einer Aufgabe sein. Satzbaupläne werden instanziiert, indem Begriffe der betrieblichen Anwendungsdomäne in die dafür vorgesehenen Platzhalter, hier *I1* und *I2* für Informationsobjekte sowie *A1* bzw. *A2* für Aufgaben, eingefügt werden. Da jedes Informationsobjekt wiederum aus anderen Informationsobjekten bestehen kann, werden jene Beziehungen benötigt, die auch zur Beschreibung der Beziehungen von Konzepten in einem UML-Klassendiagramm oder einem Entity-Relationship-Diagramm genutzt werden: Generalisierung und Komposition. Damit werden zum Ausdruck der *Beziehung zwischen Informationsobjekten* die beiden folgenden Satzbaupläne benötigt:

- *I1* ist ein *I2*
- *I1* beinhaltet *X* bis *Y I2*

Da die Informationsobjekte entweder Input- oder Output einer Funktion sind, werden hierfür zwei weitere Satzbaupläne eingeführt. Für benötigte Informationsobjekte gilt zusätzlich, dass sie optional sein können, erzeugte Informationsobjekte sind dagegen immer obligatorisch, da jede Aufgabe das Ziel hat, einen bestimmten Zustand zu erreichen. Die *Beziehungen zwischen Aufgaben und Informationsobjekten* können mit folgenden drei Satzbauplänen beschrieben werden:

- *A1* benötigt *I1*
- *A1* kann *I1* benötigen
- *A1* erzeugt *I1*

Zur Beschreibung der *Reihenfolgebeziehungen zwischen Aufgaben* werden die Sequenz sowie die optionale Sequenz benötigt. Außerdem kann jede Aufgabe in Teilaufgaben zerlegt werden, weshalb drei weitere Satzbaupläne zum Ausdruck der Beziehungen zwischen Aufgaben nötig sind:

- Nach ausführen *A1* folgt ausführen *A2*
- Nach ausführen *A1* kann ausführen *A2* folgen
- *A1* ist Teilaufgabe von *A2*

Fallstudien zur Spezifikation von fachlichen Software-Komponenten sowie Laborexperimente zeigen, dass diese acht Satzbaupläne alle Beziehungstypen zur Verfügung stellen, die zum Aufbau einer fachlichen Beschreibung der von Softwarekomponenten unterstützten betrieblichen Aufgaben und der zugehörigen Informationsobjekte benötigt werden [Zaha05, S. 42ff.].

Abbildung 1 zeigt den Zusammenhang zwischen Informationsobjekten und fachlichen Aufgaben sowie die im weiteren Verlauf zur Darstellung verwendete Notation. Diese ermöglicht die Visualisierung der in den Satzbauplänen definierten Sachverhalte. Auf der linken Seite der Abbildung finden sich die Input-Informationsobjekte in Form eines Informationsobjektbaums wieder (*Informationsobjekt 3* beinhaltet *1* bis *n* *Informationsobjekt 1*). Dies erkennt man anhand des Punkts am Kantenende bei der Aufgabe. Die gestrichelte Linie bezeichnet dabei einen Inputinformationsobjektbaum, der optional ist (*Aufgabe 1* kann *Informationsobjekt 3* benötigen). Inputinformationsobjekt 4 ist eine Generalisierung (*Informationsobjekt 2* ist ein *Informationsobjekt 4*, *Informationsobjekt 5* ist ein *Informationsobjekt 4*). Rechts im Bild ist der durch die Aufgabe erstellte Informationsobjektbaum zu sehen (Punkt am Kantenende befindet sich an der Wurzel des Informationsobjektbaums).

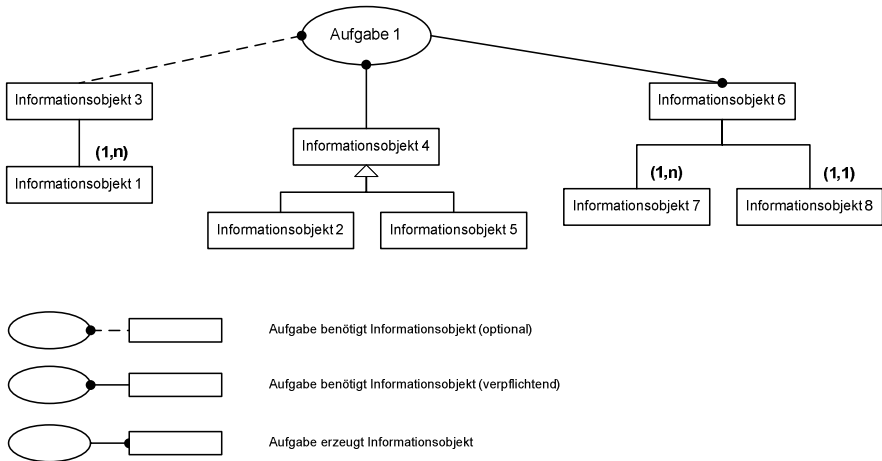


Abb. 1: Informationsobjektbäume und Aufgaben in grafischer Notation

Die für mehrdimensionale Kompatibilitätstests zu beachtenden Sachverhalte ergeben sich aus den Interdependenzen zwischen den in [Turo02] identifizierten Spezifikationsobjekten, die dort, wie oben erläutert, in Abhängigkeit ihrer Nähe zur Informationstechnik auf verschiedenen Ebenen angeordnet sind. Abbildung 2 zeigt die relevanten Beschreibungsebenen und ihre Abhängigkeiten. Auf der rechten Seite sind jene technischen Objekte der Spezifikation abgebildet, die bei Kompatibilitätstests berücksichtigt werden sollten, da sie einen direkten Bezug zu den fachlichen Aspekten der Spezifikation aufweisen. Dieser direkte Bezug ist gegeben, wenn Objekte der Spezifikation auf der fachlichen Seite direkt auf Objekte der Spezifikation auf der technischen Seite abgebildet werden können. Auf der fachlichen Seite stehen hierbei die Aufgaben- und Terminologieebene.

Auf der *Aufgabenebene* werden betriebliche Aufgaben beschrieben sowie die Kompositionsbeziehungen und Reihenfolgebeziehungen zwischen ihnen. Da die Satzbaupläne durch das Einfügen von Fachbegriffen der betrieblichen Anwendungsdomäne instanziiert werden und diese Begriffe auf *Terminologieebene* definiert werden, besteht zwischen den Ebenen eine Abhängigkeit. Eine weitere Abhängigkeit besteht innerhalb der Aufgabenebene, da sich In- und Outputbeziehungen zwischen Aufgaben und Informationsobjekten direkt auf die mögliche Reihenfolge der Aufgabenausführung auswirken. Abhängigkeiten bestehen ebenfalls auf technischer Seite zwischen Schnittstellen- und Abstimmungsebene. Mögliche Aufrufreihenfolgen der Dienstoperationen werden in der Abstimmungsebene beschrieben, wobei die Dienste selbst bereits in der Schnittstellenebene spezifiziert werden. Dort findet auch die Definition der verwendeten Datentypen statt. Die Abhängigkeit zwischen Datentypen und Diensten ist gegeben, da jeder Datentyp, der nicht ausschließlich zum Aufbau weiterer komplexer Datentypen genutzt wird, den Übergabe- oder Rückgabeparameter einer angebotenen Dienstoperation typisiert.

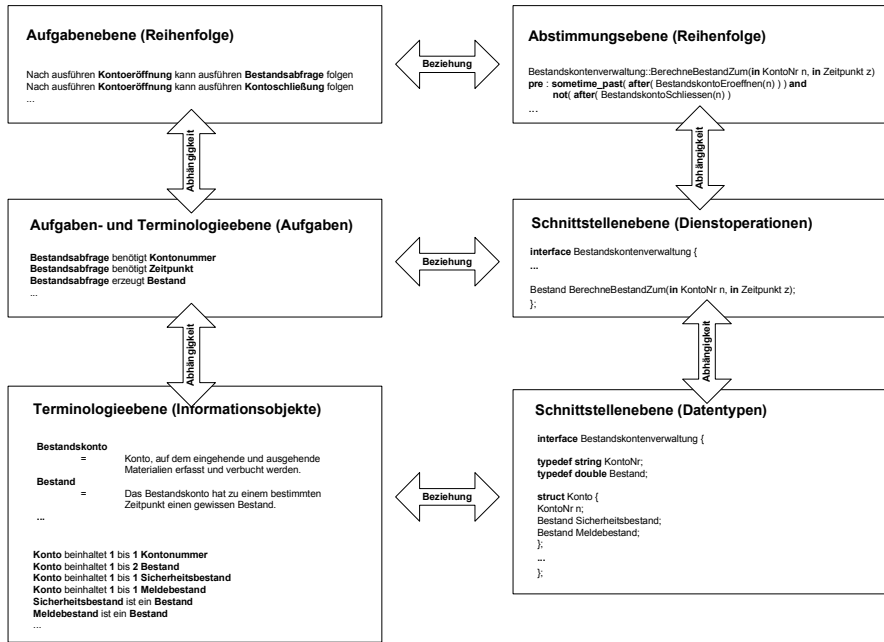


Abb. 2: Interdependenzen zwischen den Spezifikationsobjekten

Die Beziehungen zwischen fachlicher und technischer Seite ergeben sich nun aus der Abbildbarkeit von Aufgaben auf Dienstoperationen (z.B. *Bestandsabfrage* und *BerechneBestandZum*) und ebenfalls zwischen den jeweiligen Reihenfolgebeziehungen. Hierzu ist anzumerken, dass die direkte Abbildung fachlicher auf technische Aspekte nur für eine bestimmte Abstraktionsebene erfolgen kann. Beispielsweise können fachliche Aufgaben, auch wenn sie bereits als Elementaraufgabe im betriebswirtschaftlichen Sinn identifiziert wurden, noch weiter in mehrere abzuarbeitende und ggf. automatisierbare Teilaufgaben unterteilt werden (z.B. Aufgabe *Kontoöffnung* enthält Teilaufgaben *Vergabe Kontonummer*, *Festlegung Sicherheitsbestand*, *Berechnung Meldebestand*, usw.). Ist dies gegeben, muss eine Zuordnung zwischen fachlicher Aufgabe und den sie implementierenden Operationen erfolgen. Weiterhin lassen sich die in Aufgaben verwendeten Informationsobjekte entsprechend mit den in den Dienstoperationen verwendeten Datentypen in Beziehung setzen (z.B. *Kontonummer* und *KontoNr*).

### 3 Messung fachlicher Kompatibilität als Grundlage für mehrdimensionale Kompatibilitätstests

Durch die beschriebene Spezifikationsmethode ist es möglich, eine einheitliche Spezifikation sowohl der Anforderungen als auch der Dienste vorzunehmen. Darauf aufbauend lassen sich nun sowohl der Kompatibilitätsbegriff als auch die Kompatibilitätstests definieren. Auf *fachlicher Ebene* soll der Grad der Kompatibilität zweier Domänenmodelle

durch eine Kennzahl angegeben werden. Aus betriebswirtschaftlicher Sicht sind Kennzahlen verdichtete, quantitative Angaben, die es einer Organisation ermöglichen, den Erreichungsgrad gesetzter Ziele zu definieren bzw. zu messen [Hild05, 112]. Sie stellen somit wichtige Sachverhalte und die durchaus auch komplexen Strukturen und Prozesse, die damit verknüpft sind, auf einfache Weise dar [ToJa98, 10; Reic97, 19; Hein99, 392]. Das hier verfolgte Ziel ist die Messung der fachlichen Kompatibilität zwischen Spezifikationen. Die Kennzahl soll geeignet sein, bei mehreren möglichen Vergleichsspezifikationen mindestens eine Vorauswahl geeigneter Dienste treffen zu können. Es werden in der praktischen Anwendung wohl selten tatsächlich übereinstimmende Spezifikationen gefunden werden, dennoch kann bereits ein hoher Grad an Übereinstimmung ausreichen, um die Anforderungen an einen zu integrierenden Dienst zu erfüllen. Für die Definition von Kennzahlen wird gefordert, dass sie dem gesetzten Ziel entsprechen müssen [Hild05, 112], d.h. im konkreten Fall, dass die Kennzahl auch alle relevanten Einflussfaktoren enthalten muss. Die Kennzahl zur Messung der fachlichen Kompatibilität setzt sich somit aus Kennzahlen für die Übereinstimmung von Informationsobjekten und Aufgaben zusammen, da sich mit diesen die fachlichen Anforderungen beschreiben lassen. Weitere Kriterien bei der Kennzahlendefinition sind ihre Eindeutigkeit sowie Akzeptanz bei den Nutzern. Zumindest muss aber die Kennzahlendefinition aber den Nutzern bekannt und nachvollziehbar sein [Hild05, 112]. Die dargestellte deduktive Herleitung der Übereinstimmungskennzahlen anhand von Hypothesen zur fachlichen Kompatibilität dient primär der Schaffung von Bekanntheit und Nachvollziehbarkeit sowie dem Nachweis der Eindeutigkeit der Kennzahlendefinition (zwei Vergleichsspezifikationen haben bei gleichen Parametern immer dasselbe Kompatibilitätsmaß). Hierzu werden zunächst zulässige Abweichungen zwischen den zu vergleichenden Informationsobjekten bzw. Aufgaben, d.h. den Elementen des Domänenmodells, vorgestellt und ausgehend von den Informationsobjekten über die Aufgaben bis zur gesamten Spezifikation die Berechnung des Kompatibilitätsmaßes dargestellt.

### 3.1 Zuordnung von Informationsobjekten

Grundsätzlich wird davon ausgegangen, dass die Vergleichsspezifikation alle in der Suchspezifikation gestellten Anforderungen erfüllen muss, jedoch kann zur Unterstützung einer fachlichen Aufgabe ggf. auch eine teilweise Übereinstimmung mit der Vergleichsspezifikation ausreichend sein (z.B. weil diese eine wichtige Aufgabe jedoch nicht alle Aufgaben unterstützt). Bei der Betrachtung von Informationsobjekten gilt allerdings die oben genannte Prämisse, da optionale Informationsobjekte im Gegensatz zu den Aufgaben nicht unabhängig betrachtet werden können. Außerdem können optionale Informationsobjekte als solche kenntlich gemacht werden. Es lassen sich somit an die Zuordnung eines Informationsobjekts aus der Suchspezifikation zu einem Informationsobjekt aus der Vergleichsspezifikation folgende zwei Bedingungen knüpfen:

1. Jedem Informationsobjekt auf jeder Ebene des Informationsobjektbaums der Suchspezifikation muss ein Informationsobjekt der entsprechenden Ebene des Informationsobjektbaums der Vergleichsspezifikation zugeordnet werden, wobei beide Informationsobjekte denselben Namen haben müssen.



2. Die Wertebereiche der Kardinalitäten der Suchspezifikation müssen innerhalb der entsprechenden Wertebereiche der Kardinalitäten der Vergleichsspezifikation liegen.

Die durch diese Bedingungen möglichen auftretenden Abweichungen zwischen zwei Informationsobjekten lassen sich in zwei Kategorien einteilen: Namensabweichungen und Strukturabweichungen.

Bei einer *Namensabweichung* müssen Abweichungen zwischen zwei komplexen Informationsobjekten und zwischen zwei elementaren Informationsobjekten unterschieden werden, da letztere durch einen elementaren Typ gekennzeichnet sind. Bei diesen Elementen kann auch bei abweichenden Namen eine vollständige Abdeckung der Anforderungen erreicht werden, wenn die Elementartypen kompatibel sind (z.B. kann eine ganze Zahl durch eine Gleitkommazahl dargestellt werden). Auch bei Namen komplexer Informationsobjekte ist eine Relativierung einer Namensabweichung möglich, da unterschiedliche Schreibweisen berücksichtigt werden können, wenn Techniken des Information Retrieval (vgl. [BaRi99]) oder ein manuelles Namensmapping angewendet werden, um Abweichungen der Schreibweise von Namen zuzulassen. In diesem Fall müsste vor dem Vergleich zunächst noch eine Zuordnung von Namen erfolgen. Aus Gründen der Komplexitätsreduktion werden diese beiden Möglichkeiten der Präzisierung von Namensabweichungen nicht im Kompatibilitätsmaß berücksichtigt. Jedoch wird angenommen, dass im Vorfeld der Kennzahlbestimmung ein mögliches Namensmapping bzw. eine Namenspräzisierung stattfindet.

*Strukturabweichungen* können zweierlei Gründe haben: zum einen können die Wertebereiche der Kardinalitäten abweichen, zum anderen können die Zuordnungen aufgrund fehlender Informationsobjekte nicht vollständig sein. Bei den Wertebereichen lassen sich, ähnlich der Kompatibilität von Elementartypen, zulässige Zuordnungen finden. Bei einer Strukturabweichung, die durch das Fehlen eines Informationsobjekts verursacht wird, kann eine Zuordnung erfolgen, falls dieses in der Suchspezifikation fehlt und durch das Hinzufügen eines virtuellen Informationsobjekts ersetzt werden kann, d.h. falls dieses zur Aufgabenerfüllung nicht zwingend notwendig ist. Diese Abweichung muss jedoch bei der Berechnung der Übereinstimmungskennzahl berücksichtigt werden. Hierzu wird das hinzugefügte Informationsobjekt mit der Kardinalität  $(0,0)$  versehen, was die Semantik des Informationsobjekts jedoch nicht verändert. Ein zweiter Fall bei dieser Art von Strukturabweichung betrifft die Anzahl der jeweiligen Abstraktionsebenen in den beiden Spezifikationen. Es kann vorkommen, dass Informationsobjektbäume eigentlich aufeinander abgebildet werden könnten, wenn nicht in einer der Spezifikationen eine Abstraktionsebene vorhanden wäre, die in der anderen Spezifikation fehlt. Im Rahmen der Berechnung der Übereinstimmungskennzahl wird diese Abweichung berücksichtigt, indem Hilfs-Informationsobjekte an der Stelle eingefügt werden, an der eine Abstraktionsebene fehlt. Die neu einzufügenden Kardinalitäten für die Beziehung zu dem neuen Informationsobjekt, das ohne Namen eingefügt wird, werden soweit an das Pendant in der anderen Spezifikation angepasst, wie es die Kardinalitäten der Informationsobjekte der darunter liegenden Ebene, die jetzt eine Ebene tiefer notiert sind, zulassen.

Weiterhin muss bei der Berechnung der Übereinstimmungskennzahl zweier Informationsobjekte berücksichtigt werden, dass durch Vererbungsbeziehungen in einer Spezifi-

kation mehrere Spezifikationen entstehen können, die alle iterativ überprüft werden müssen. Ein Beispiel wären zwei Informationsobjekte, die eine Vererbungsbeziehung zu einem übergeordneten Informationsobjekt haben. Dann würden dadurch drei mögliche Spezifikationen entstehen, da an der Stelle das vererbende Informationsobjekt selbst oder eines der beiden spezialisierten Informationsobjekte stehen könnte.

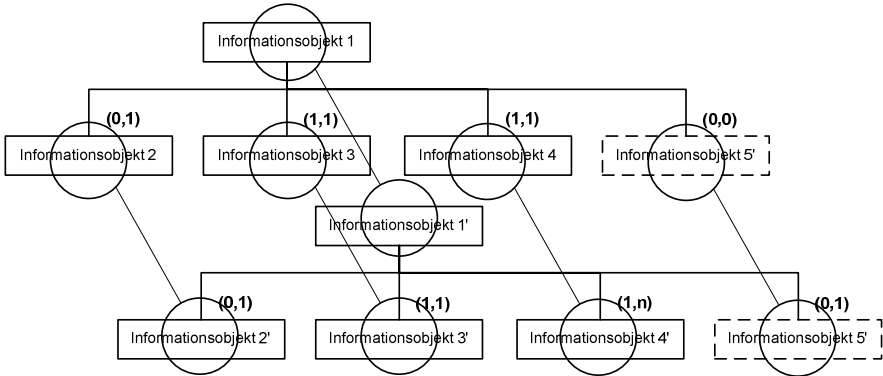


Abb. 3: Zuordnung von Informationsobjekten

Ein Beispiel für die Erfüllung der geschilderten Zuordnungskriterien – abgesehen von der Namensabweichung – ist in Abbildung 3 dargestellt, wobei in dieser Grafik die Suchspezifikation oben und die Vergleichspezifikation unten notiert ist. Zur Berechnung der Übereinstimmungskennzahl, die alle oben beschriebenen Zuordnungsmöglichkeiten bei Abweichungen berücksichtigt, wird Funktion 1 verwendet. Die Übereinstimmungskennzahl ergibt sich durch eine rekursive Anwendung dieser Funktion über die durch den Ebenenparameter  $e$  angegebene Anzahl zu betrachtender Ebenen des Informationsobjektbaums und kann Werte zwischen 0 und 1 annehmen. Der Parameter  $e$  dient hierbei lediglich der Beschränkung der Laufzeit des Kompatibilitätsalgorithmus bei sehr großen Spezifikationen, und ist nur bei Spezifikationen sinnvoll, die Informationsobjektbäume enthalten, die mehr als 10 Ebenen tief sind.

$$\ddot{u}_{z_i}(\alpha, \beta, e) = \frac{\alpha \cdot \ddot{u}_{N_i} + \left( \frac{\sum_{i=1}^a \ddot{u}_{K_i}(\beta) \cdot \ddot{u}_{z_{ii}}(\alpha, \beta, e-1)}{a} \right)}{1 + \alpha}, \quad 0 \leq \alpha \leq 1, \quad 0 \leq \beta \leq 1, \quad e \geq 1$$

Fkt. 1: Übereinstimmung von Informationsobjekten

Der Wert von  $\ddot{u}_{z_i}$  drückt aus, inwieweit zwei Informationsobjektbäume unter Berücksichtigung von Strukturabweichungen und Namensabweichungen übereinstimmen.  $\ddot{u}_{N_i}$  gibt hierbei die Namensübereinstimmung an, die den Wert 1 hat, wenn die beiden Informationsobjekte denselben Namen haben und 0, wenn die beiden Namen unterschiedlich sind. Diese Namensübereinstimmung wird mit  $\alpha$  gewichtet, um auszudrücken, in welchem Maße die Namensübereinstimmung bei der Kompatibilitätsbetrachtung eine Rolle spielen soll. Der Anteil der Übereinstimmung, der sich für die betrachtete Ebene

aus den Zuordnungen auf der direkt untergeordneten Ebene des Informationsobjektbaums ergibt, wird folgendermaßen berechnet: Die Übereinstimmungskennzahl jedes assoziierten Informationsobjekts  $\ddot{u}_{z_i}$  wird mit der Übereinstimmungskennzahl des Wertebereichs der Kardinalitäten  $\ddot{u}_{k_i}$  multipliziert, die 0 bei unterschiedlichen, 1 bei identischen und  $(1-\beta)$  bei überlappenden Wertebereichen ist. Überlappend bedeutet hierbei, dass der Wertebereich der Kardinalität des untergeordneten Informationsobjekts in der Suchspezifikation innerhalb des Wertebereichs der Kardinalität des entsprechenden Informationsobjekts in der Vergleichsspezifikation liegt. Bei eingefügten Hilfsobjekten wird ebenfalls der Wert  $(1-\beta)$  für  $\ddot{u}_{k_i}$  verwendet, um die dadurch entstandene Strukturabweichung zu bewerten. Schließlich wird das arithmetische Mittel über die Anzahl  $a$  der assoziierten Informationsobjekte der Suchspezifikation berechnet. Nach der Addition mit der oben bereits erwähnten Namensübereinstimmung wird die Kennzahl mittels Division durch  $(1+\alpha)$  auf Werte zwischen 0 und 1 normiert. Für die Wahl der Werte für die Parameter  $\alpha$  und  $\beta$  gilt, dass die Extreme 0 oder 1 in den seltensten Fällen sinnvoll sein dürften. So hätte z.B. die Gewichtung der Namensabweichung mit 1 zum Ergebnis, dass zwei Informationsobjekte, deren Bestandteile auf der einen Seite in Einzahl und auf der anderen Seite in Mehrzahl beschrieben sind, inkompatibel sind. Damit ist die Wahl der Werte für diesen Parameter abhängig davon, inwieweit Unterschiede in der Schreibweise bei der Durchführung des Algorithmus berücksichtigt werden beziehungsweise wie stark die zugehörige Software bezüglich der Namensgebungen parametrisiert werden kann. Für die Wahl des Wertes des Parameters  $\beta$  gilt ebenfalls, dass ein Wert zwischen 0 und 1 gewählt werden sollte, allerdings ist hier der Grund die Vagheit der Beschreibung der Struktur von Objekten der betrieblichen Anwendungsdomäne. Zwar existieren Fälle, in denen die realen Erfordernisse zwingend den Aufbau von Informationsobjekten vorschreiben, doch dürfte in den meisten Fällen ein erheblicher Gestaltungsspielraum für deren Beschreibung vorhanden sein. Für zwei Informationsobjekte, die jeweils vier Kindobjekte haben und bei denen keine Namensübereinstimmung existiert (Beispiele für solche Informationsobjekte finden sich in Abbildung 3) ergeben für verschiedene Werte von  $\alpha$  und  $\beta$  und unterschiedliche Übereinstimmungen in den Kardinalitäten der Kindobjekte folgende  $\ddot{u}_{z_i}$ -Werte:

– $\alpha$	– $\beta$	– Übereinstimmung der Kardinalität	– $\ddot{u}_{z_i}$
– 0,2	– 0,7	– bei 2 von 4 Kindobjekten	– 0,45
– 0,2	– 0,5	– bei 2 von 4 Kindobjekten	– 0,52
– 0,2	– 0,7	– bei 3 von 4 Kindobjekten	– 0,57

Tab. 1: Beispiele für Übereinstimmungskennzahlen

Anhand von Tabelle 1 erkennt man, dass bei gleich bleibenden  $\alpha$  und  $\beta$  – wie gefordert – die Kennzahl bei zunehmender Übereinstimmung ansteigt. Ein niedrigeres  $\beta$  und damit eine niedrigere Gewichtung von Abweichungen in der Kardinalität wirkt sich ebenfalls positiv auf die Kennzahl aus.

### 3.2 Zuordnung von Aufgaben

Die vorgestellte Übereinstimmungskennzahl von Informationsobjekten wird im Rahmen der Berechnung der Übereinstimmungskennzahl von Aufgaben verwendet. Für eine Zuordnung von Aufgaben werden ebenfalls Bedingungen und zulässige Abweichungen definiert, die im Folgenden vorgestellt werden (eine mögliche Namensabweichung wird hier ebenfalls nicht betrachtet). Bedingungen für eine Zuordnung sind:

1. Jedem zur Durchführung der Aufgabe unbedingt erforderlichen Informationsobjekt der Suchspezifikation muss ein unbedingt oder optional erforderliches Informationsobjekt der Vergleichsspezifikation zugeordnet werden können.
2. Jedem zur Durchführung der Aufgabe optional erforderlichen Informationsobjekt der Suchspezifikation muss ein optionales Informationsobjekt der Vergleichsspezifikation zugeordnet werden können und die Ergebnisse der beiden Aufgaben müssen einander entsprechen.
3. Für die vollständige Zuordnung zweier Aufgaben muss sichergestellt sein, dass kein zwingend erforderliches Informationsobjekt zur Durchführung einer Aufgabe aus der Vergleichsspezifikation unzugeordnet bleibt.

Ein Beispiel für eine Zuordnung, die diese Kriterien erfüllt, ist in Abbildung 4 dargestellt. Die Suchspezifikation findet sich wiederum oben und die Vergleichsspezifikation unten.

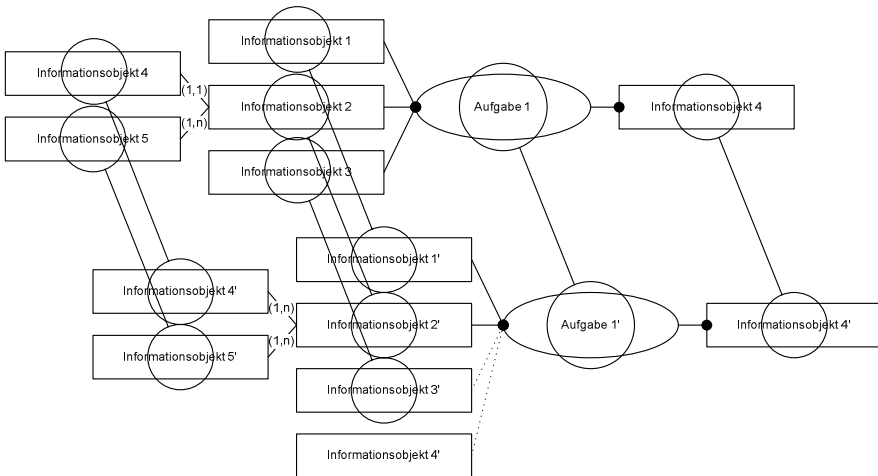


Abb. 4: Abbildung zweier Domänenmodelle

Neben der Namensabweichung können also bei der Zuordnung zweier Aufgaben Abweichungen dadurch entstehen, dass Input- oder Output-Informationsobjekte nicht nach den genannten Kriterien zugeordnet werden können. Diese Abweichungen können darin begründet sein, dass in einer Spezifikation Abstraktionsebenen vorhanden sind, die in der anderen Spezifikation fehlen. Deshalb muss im Rahmen der Berechnung einer Über-

einstimmungskennzahl für Aufgaben auch die Zusammenfassung mehrerer Aufgaben betrachtet werden. Eine *Zusammenfassung von Aufgaben* ist unter zwei Bedingungen möglich:

1. Output-Informationsobjekte, die kein Ergebnis der Gesamtaufgabe darstellen, müssen auf die Input-Informationsobjekte, die keinen Input der Gesamtaufgabe darstellen, abgebildet werden können.
2. Es dürfen nur Zusammenfassungen vorgenommen werden, die nicht im Widerspruch zu explizit spezifizierten Reihenfolgebeziehungen stehen. Die Voraussetzung für die Zusammenfassung von Aufgaben bezüglich der Reihenfolgebeziehungen ist wie folgt definiert: hat eine Aufgabe eine Reihenfolgebeziehung zu anderen Aufgaben, so kann sie nur zusammengefasst werden, wenn mindestens alle diese Aufgaben in der Zusammenfassung enthalten sind, da die Reihenfolgebeziehungen zwischen Aufgaben nur innerhalb einer Abstraktionsebene zulässig sind und bei der Verletzung dieser Regel, explizit definierte Reihenfolgebeziehungen verloren gehen.

Damit kann die Funktion für die Berechnung der Übereinstimmungskennzahl für Aufgaben wie folgt angegeben werden:

$$\ddot{u}_{Z_A}(\alpha, \beta, e) = \frac{\alpha \cdot \ddot{u}_{N_A} + \left( \frac{\ddot{u}_{Z_{oi}}(\alpha, \beta, e) + \sum_{i=1}^b \ddot{u}_{Z_{ii}}(\alpha, \beta, e)}{b+1} \right)}{1 + \alpha}, 0 \leq \alpha \leq 1, 0 \leq \beta \leq 1, e \geq 1$$

Fkt. 2: Übereinstimmung von Aufgaben

Der Funktion zur Berechnung der Übereinstimmungskennzahl werden wiederum die drei Parameter  $\alpha$ ,  $\beta$  und  $e$  übergeben. In der Funktion selbst wird jedoch nur der Parameter  $\alpha$  zur Gewichtung der Namensabweichung zweier Aufgaben  $\ddot{u}_{N_A}$ , die entweder den Wert 0 oder den Wert 1 annehmen kann, berücksichtigt, da eine Zuordnung der Aufgaben nur auf der niedrigsten Abstraktionsebene durchgeführt wird. Der Parameter zur Gewichtung von Strukturabweichungen  $\beta$  sowie der Ebenenparameter  $e$  sind mit denen bei der Berechnung der Übereinstimmungskennzahl für Informationsobjekte identisch.  $\ddot{u}_{Z_{ii}}$  sind die jeweiligen Übereinstimmungskennzahlen der Inputinformationsobjekte und  $\ddot{u}_{Z_{oi}}$  die Übereinstimmungskennzahl des Outputinformationsobjekts. Diese werden gemittelt, wobei  $b$  die Anzahl der Inputinformationsobjekte angibt. Die gesamte Übereinstimmungskennzahl für die betrachtete Aufgabe wird wiederum über eine Division durch  $1 + \alpha$  auf 1 normiert. Da im Rahmen der fachlichen Spezifikation einer Software-Komponente typischerweise mehrere Aufgaben beschrieben werden, wird abschließend eine Kennzahl eingeführt, die den Grad der Übereinstimmung zweier Gesamtspezifikationen auf fachlicher Ebene ausdrückt:

$$\ddot{u}_{Z_F}(\alpha, \beta, e) = \frac{1}{c} \cdot \sum_{i=1}^c \ddot{u}_{Z_{A_i}}(\alpha, \beta, e)$$

Fkt. 3: Übereinstimmung von Spezifikationen

$\ddot{u}_{Z_F}$  ist definiert als das arithmetische Mittel über alle Übereinstimmungskennzahlen  $\ddot{u}_{Z_{A_i}}$  der in der Suchspezifikation enthaltenen Aufgaben und stellt somit das fachliche Kompatibilitätsmaß dar.

## 4 Mehrdimensionale Kompatibilitätstests

Im Folgenden wird ein Vorgehensmodell für die Durchführung von mehrdimensionalen Kompatibilitätstests vorgestellt, bei dem das in Abschnitt 3 definierte Kompatibilitätsmaß für den fachlichen Test genutzt sowie auch die technische Kompatibilitätsprüfung vorgenommen wird. Die ausführliche Beschreibung der Kompatibilitätstests findet sich in [Zaha05]. Abbildung 5 zeigt den generellen Ablauf der Tests auf den betrachteten Ebenen nach der Überführung der fachnormsprachlichen Notation, d.h. der Überführung der Satzbaupläne, in eine Ontologiesprache. Für den fachlichen Kompatibilitätstest wurde dabei die Web Ontology Language (OWL) [DeSc03] aufgrund der zu erwartenden W3C-Standardisierung und somit ihrem Verbreitungspotenzial als die am besten geeignete Sprache identifiziert, um eine automatisierte Weiterverarbeitung zu erleichtern [Zaha05, S. 42ff.]. Mit der Überführung der Satzbaupläne in OWL erfolgt gleichzeitig die Zusammenführung der Beschreibung der von einander abhängigen Terminologie- und Aufgabenebene. Hier kann ebenfalls eine ggf. notwendige Namenspräzisierung stattfinden.

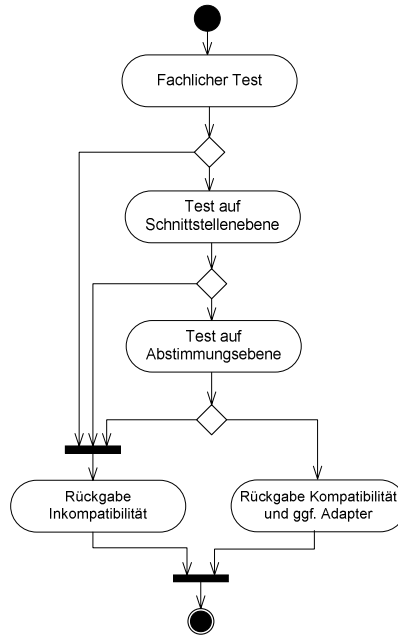


Abb. 5: Ablauf des mehrdimensionalen Kompatibilitätstests

Da eine Kompatibilität zweier Spezifikationen nur dann gewährleistet ist, wenn neben der Abbildbarkeit von Spezifikationen einzelner Ebenen aufeinander auch alle Abhängigkeiten zwischen den Spezifikationsebenen berücksichtigt werden, muss ein Algorithmus dies in seinem Ablauf berücksichtigen. Der Kompatibilitätsalgorithmus beginnt daher mit dem Test auf fachlicher Ebene, da nur bei einer Übereinstimmung der Domänenmodelle eine weitere Überprüfung der technischen Eigenschaften sinnvoll ist.

Im Erfolgsfall wird zunächst mit dem Test auf Schnittstellenebene und dann mit dem Test auf Abstimmungsebene fortgefahren. Durch die gewählte Reihenfolge ist es möglich, das Ergebnis des Schnittstellentests während der Ausführung des Tests auf Abstimmungsebene zu berücksichtigen, da dort lediglich die Reihenfolgenbeziehungen anhand von Dienstnamen und nicht mehr anhand von Signaturen abgeglichen werden. Der Algorithmus für den Test auf Abstimmungsebene kann damit vollständig abgearbeitet werden, ohne dass widersprüchliche Resultate entstehen. Verlieft auch der Test auf Abstimmungsebene erfolgreich, wird als Ergebnis Kompatibilität und ein Adapter für die Schnittstellenspezifikationen zurückgegeben.

Ergebnis des *fachlichen Tests* sind die mit Hilfe des Kompatibilitätsmaßes bewerteten Kombinationen von Zuordnungen zwischen Such- und Vergleichsspezifikation. Zu diesem Zweck wird zunächst eine Liste von gesuchten Aufgaben aus der Suchspezifikation erstellt, die sich nicht in weitere Aufgaben unterteilen lassen. Für jede dieser Aufgaben wird eine Menge von Aufgaben der Vergleichsspezifikation gesucht, für die eine Zuordnung der Input- und Output-Informationsobjekte potentiell möglich ist. Dies geschieht über den Abgleich der Informationsobjektbaumstrukturen mit Hilfe so genannter Cha-

rakteristiken. Diese geben pro Knoten im Informationsobjektbaum an, welche und wie viele Blätter inklusive der dazugehörigen Elementartypen darunter liegen. Diese Suche nach Kandidaten berücksichtigt auch Informationsobjektbäume, die durch Aufgabenzusammenfassungen entstehen können. Die Anzahl der Kandidaten kann recht umfangreich sein, da Zuordnungen lediglich auf Basis der Charakteristiken, d.h. der unterhalb eines Knotens befindlichen Elementartypen, stattfinden. Beispielsweise ergeben sich für zwei Spezifikationen mit drei identischen, als Zeichenketten dargestellten Informationsobjekte *Straße*, *Hausnummer* und *Ort* neun Zuordnungsmöglichkeiten. Für die identifizierten Kandidaten wird im nächsten Schritt versucht, eine komplette Zuordnung zu finden. In diesem Schritt werden die in Abschnitt 3 erläuterten zulässigen Abweichungen berücksichtigt und die Informationsobjekte gegebenenfalls durch das Einfügen virtueller Ebenen bzw. virtueller Informationsobjekte ergänzt. Ebenso werden alle Kombinationen, die sich aus der Abstraktion von Informationsobjekten im Sinne einer Zusammenfassung von Informationsobjekten über Ebenen ergeben, betrachtet. Aus den möglichen Zuordnungen der Informationsobjekte aus der Suchspezifikation zur Vergleichsspezifikation werden entsprechend der Input-Output-Bedingungen mögliche Kombinationen für eine Aufgabenzuordnung vorgenommen. Hierbei sind ebenfalls Zusammenfassungen von Aufgaben der Suchspezifikation zulässig, um zu einer Zuordnung zu gelangen. Jede dieser möglichen Aufgabenzuordnungen wird mit Hilfe der vorgestellten Kennzahl bewertet und das Ergebnis zurückgeliefert. Der fachliche Test ist damit beendet und es kann, sofern mindestens eine Zuordnung gefunden wurde, welche die geforderte und zuvor festgelegte Mindestkompatibilität aufweist, mit dem Test auf Schnittstellenebene fortgefahren werden.

Wie aus der obigen Beschreibung des Vorgehensmodells zur Kompatibilitätsprüfung hervorgeht, liefert der *Test auf Schnittstellenebene* neben der Aussage über die Kompatibilität zweier Softwarekomponenten auf dieser Ebene auch einen Adapter, der festlegt, wie die einzelnen Bestandteile der Schnittstellen aufeinander abgebildet werden können, falls die Schnittstellen kompatibel sind (vgl. dazu auch [ZaGG04]). Die Kompatibilitätsprüfung basiert auf den IDL-Konstrukt Schnittstelle, Dienst, Datentyp und Ausnahme. Für die Schnittstellen werden drei Arten von Kompatibilität unterschieden: exakte Kompatibilität, Spezialisierung und Generalisierung. Es gilt:

- Zwei Schnittstellen  $S$  und  $S'$  sind exakt kompatibel, wenn sie exakt dieselben Dienste in Abhängigkeit von den zugehörigen Datentypen anbieten und nachfragen (Notation:  $S = S'$ ).
- Eine Schnittstelle  $S$  ist spezieller kompatibel zu  $S'$ , wenn  $S$  mehr Dienste und Datentypen oder speziellere Dienste und Datentypen als  $S'$  anbietet oder nachfragt (Notation:  $S < S'$ ).
- Eine Schnittstelle  $S$  ist allgemeiner kompatibel zu  $S'$ , wenn  $S'$  spezieller kompatibel zu  $S$  ist (Notation:  $S > S'$ ).

Bei Diensten und Datentypen werden ebenfalls die drei oben genannten Arten von Kompatibilität unterschieden. Zwei Dienste sind exakt kompatibel, wenn ihre Parameter und Rückgabewerte exakt kompatibel sind. Voraussetzung für speziellere Kompatibilität ist die exakte Kompatibilität der Ausnahmen und Übergabeparameter sowie die speziellere Kompatibilität der Rückgabewerte – allgemeinere Kompatibilität ist entsprechend defi-



niert. Die Kompatibilität zweier Datentypen ist davon abhängig, ob es sich um einen komplexen oder elementaren Datentyp handelt. Zwei komplexe Datentypen sind exakt kompatibel, wenn die Anzahlen ihrer zugeordneten Datentypen identisch sind und diese selbst wieder exakt kompatibel sind. Ein komplexer Datentyp ist spezieller kompatibel zu einem anderen komplexen Datentyp, wenn er aus mehr Datentypen besteht und diese spezieller oder exakt kompatibel abgebildet werden können. Die Art der Kompatibilität elementarer Datentypen kann für die in der IDL-Spezifikation vorgesehenen 13 Typen durch vollständigen, gegenseitigen Vergleich bestimmt werden. Der Kompatibilitätsalgorithmus prüft nun die Zuordnung der spezifizierten Dienste (Parameteranzahl, Rückgabe- und Übergabeparameter) und, falls diese Abbildung erfolgreich durchgeführt werden konnte, die Zuordnung der Datentypen, die nicht schon im Rahmen der Abbildung der Dienste erfolgt ist. Ein Beispiel für einen Dienstevergleich zeigt Abbildung 6. Der Vergleich der Operationen *BerechneBestandZum* und *BestandAm* führt zum Ergebnis einer spezieller kompatiblen Schnittstelle in der Suchspezifikation *BerechneBestandZum*, da alle Inputparameter zugeordnet werden konnten, aber der Datentyp *Konto* spezieller ist als *KontoNr*, da diese in dessen Typdefinition enthalten ist und *Konto* noch mehr Informationen speichern kann. Alle beim Vergleich von Diensten bzw. Datentypen gefundenen Abbildungen werden in einer Adapterliste gesammelt und zu einem Gesamt-Adapter integriert. Die Generierungsregeln für Adapter sowie die Vorgehensweise bei ihrer Integration finden sich in [Zaha05, 141ff.].

```

interface Bestandskontenverwaltung {
    typedef string KontoNr;
    typedef double Bestand;
    typedef string Zeitpunkt

    struct Konto {
        KontoNr n;
        Bestand Sicherheitsbestand;
        Bestand Meldebestand;
    };

    Bestand BerechneBestandZum(in Konto n, in Zeitpunkt z);
    boolean Aktiv(in KontoNr nr);
};

interface Kontenverwaltung {
    typedef string KontoNr;
    typedef double Bestand;
    typedef string Zeitpunkt

    Bestand BestandAm(in KontoNr nr, in Zeitpunkt t);
    boolean IstAktiv(in KontoNr nr);
};
    
```

Dienstmatrix	BestandZum	IstAktiv
BerechneBestandZum	{<=<(BerechneBestandZum, BestandAm),(0,0),(1,1)}, <=<(Konto, KontoNr),(1,1)}, =<=<(Zeitpunkt, Zeitpunkt)}	{}
Aktiv	{}	{=<=<(Aktiv, IstAktiv),(0,0)}

Abb. 6: Beispiel für einen Dienstevergleich bei Schnittstellen

Der *Test auf Abstimmungsebene* wird nur ausgeführt, wenn die Schnittstellen der spezifizierten Komponenten – gegebenenfalls unter Verwendung eines Adapters – kompatibel sind. Jedes Spezifikationsartefakt auf Abstimmungsebene stellt immer entweder eine Vor- oder eine Nachbedingung einer bestimmten Dienstoperation dar. Um die Äquivalenz zweier Vor- oder Nachbedingungen zu überprüfen, müssen zunächst für jede spezifizierte Reihenfolgebeziehung der einen Spezifikation alle zu überprüfenden Reihenfolge-

genbeziehungen der zweiten Spezifikation identifiziert werden. Diese zulässigen Kombinationen lassen sich aus dem im Rahmen des Tests auf Schnittstellenebene erzeugten Adapter ableiten. Da jedes Spezifikationsartefakt auf Abstimmungsebene einer bestimmten Dienstoperation zugeordnet ist, müssen für diese Dienstoperationen die kompatiblen Dienstoperationen der Vergleichsspezifikation aus dem Adapter abgelesen werden. Alle für diese Dienstoperationen spezifizierten Reihenfolgebeziehungen der Vergleichsspezifikation stellen jene Menge dar, für die eine Prüfung auf logische Äquivalenz jeweils für Vor- und Nachbedingungen durchzuführen ist. Die Prüfung erfolgt durch Anwendung von Umformungsregeln für temporallogische Ausdrücke [ZaAl06]. Lässt sich eine Suchspezifikation durch Umformungen in die Vergleichsspezifikation überführen, so werden diese als logisch äquivalent bezeichnet. Werden diese Umformungen für mindestens eine Kombination von Dienstoperationen, für die eine Reihenfolgebeziehung spezifiziert wurde, erfolgreich durchgeführt, ist die Kompatibilität der beiden Spezifikationen auf Abstimmungsebene – in Abhängigkeit von der Schnittstellenkompatibilität – festgestellt.

## 5 Zusammenfassung und Ausblick

Im Rahmen dieses Beitrags wurde die Notwendigkeit zum mehrdimensionalen Test auf Kompatibilität bei der Suche von Diensten herausgestellt und ein Kompatibilitätsmaß auf Basis einer fachlichen Spezifikation der Anwendungsdomäne hergeleitet. Schließlich wurde gezeigt, wie ein Kompatibilitätstest für fachliche und technische Ebenen aufgebaut ist und wie das Kompatibilitätsmaß zur Bewertung von Alternativen herangezogen werden kann. Dieser mehrdimensionale Kompatibilitätstest eignet sich sowohl bei der kurzfristigen Einbindung externer Funktionalität in Ad-hoc-Szenarien als auch bei der langfristigen Kopplung heterogener Anwendungssysteme. In beiden Fällen wird eine effiziente Methode zur Bestimmung von Kompatibilität benötigt, um einerseits eine automatisierte Einbindung zu ermöglichen und andererseits den Aufwand für eine mögliche Integrationsaufgabe beim Aufbau serviceorientierter Systeme abzuschätzen.

Zur Prüfung der Anwendbarkeit und Validierung wurde der vorgestellte Kompatibilitätstest in Form eines integrierten Spezifikations- und Testwerkzeugs prototypisch implementiert und im Rahmen einer Fallstudie im Bereich der Ressourcenplanung angewendet [Zaha05, 121ff.]. Das Kompatibilitätsmaß zeigte sich hierbei als für die fachliche Bewertung von Spezifikationen geeignet. Es wurde jedoch noch ein Verbesserungspotenzial in der Vorgehensweise für die mehrdimensionale Spezifikation der Anforderungen und Dienste festgestellt, da der Prozess zur Beschreibung der Software bzw. zur Erstellung der Anforderungsspezifikation sehr aufwändig ist. Gerade für die fachliche Spezifikation der Anwendungsdomäne gibt es bisher kaum Möglichkeiten, den Aufwand durch den Einsatz geeigneter Methoden zu verringern. Ebenso bieten sowohl die Überführung existierender natürlichsprachiger Spezifikationen in formale Beschreibungen als auch die Generierung neuer Spezifikationsartefakte aus unvollständigen Domänenbeschreibungen große Verbesserungspotenziale im Bezug auf Effizienz.

## 6 Literaturverzeichnis

- [ABBK02] Atkinson, C. et. al.: Component-based Product Line Engineering with UML. Addison Wesley, Reading 2002.
- [BaRi99] Baeza-Yates, R.; Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Boston, 1999.
- [CCCD01] Cauldwell, P. et. al.: XML Web Services. Wrox Press Ltd., Birmingham, 2001.
- [DeSc03] Dean, M.; Schreiber, G. (Hrsg.): OWL Web Ontology Language Reference. W3C Recommendation. <http://www.w3.org/TR/owl-ref/>, Abruf 2005-02-09.
- [Fran94] Frank, U.: Multiperspektivische Unternehmensmodellierung. Theoretischer Hintergrund und Entwurf einer objektorientierten Entwicklungsumgebung. Oldenbourg Verlag, München, 1994.
- [GaAO95] Garlan, D.; Allen, R.; Ockerbloom, J.: Architecture Mismatch: Why Reuse is so Hard. In: IEEE Software 12 (1995) 6; S. 17-26.
- [Hein99] Heinrich, L. J.: Informationsmanagement: Planung, Überwachung und Steuerung der Informationsinfrastruktur. 6. Auflage, Oldenbourg Verlag, München, 1999.
- [Hild05] Hildebrand, K.: Glossar zum Schwerpunktthema Supply Chain Management. In: HMD – Praxis der Wirtschaftsinformatik 243 (2005).
- [HoWo04] Hohpe, G.; Woolf, B.: Enterprise integration patterns. Addison-Wesley, Boston, 2004.
- [Kaib02] Kaib, M.: Enterprise Application Integration. Grundlagen, Integrationsprodukte, Anwendungsbeispiele. Deutscher Universitäts-Verlag, 2002.
- [KrTu02] Krammer, A.; Turowski, K.: Spezifikationsbedarf von Web-Services. In (E. Ortner; S. Overhage) (Hrsg.): 1. Workshop Entwicklung von Anwendungen auf der Basis der XML Web-Service Technologie. Darmstadt 2002; S. 1-16.
- [OMG01] OMG (Hrsg.): OMG Unified Modeling Language Specification, Version 1.4, September 2001. Needham, 2001.
- [OMG02] OMG (Hrsg.): The Common Object Request Broker: Architecture and Specification: Version 3.0, July 2002. OMG, Framingham, 2002.
- [Ortn97] Ortner, E.: Methodenneutraler Fachentwurf: Zu den Grundlagen einer anwendungsorientierten Informatik. Teubner, Stuttgart, 1997.
- [RaTu01] Rautenstrauch, C.; Turowski, K.: Common Business Component Model (COBCOM): Generelles Modell komponentenbasierter Anwendungssysteme. In (Buhl, H.U.; Huther, A.; Reitwiesner, B.) (Hrsg.): Information Age Economy: 5. Internationale Tagung Wirtschaftsinformatik 2001, Augsburg, 19.09.-21.09.2001. Augsburg, 2001; S. 681-695.
- [Reic97] Reichmann, T.: Controlling mit Kennzahlen und Managementberichten. Vahlen, München, 1997.
- [ToJa98] Totok, A.; Jaworski, R.: Modellierung von multidimensionalen Datenstrukturen mit ADAPT. Technische Universität Braunschweig, Braunschweig, 1998.

- [Turo02] Turowski, K. (Hrsg.): Vereinheitlichte Spezifikation von Fachkomponenten: Memorandum des Arbeitskreises 5.10.3 Komponentenorientierte betriebliche Anwendungssysteme, Februar 2002. Universität Augsburg, Augsburg, 2002.
- [Wied95] Wiederhold, G.: Mediation in information systems. In: ACM Computing Surveys 27 (1995) 2; S. 265–267.
- [Wöhe96] Wöhe, G.: Einführung in die Allgemeine Betriebswirtschaftslehre. Verlag Franz Vahlen, München, 1996.
- [YeSt97] Yellin, D. M.; Strom, R. E.: Protocol Specifications and Component Adaptors. In: ACM Transactions on Programming Languages and Systems 19 (1997) 2; S. 292-333.
- [ZaAl06] Zaha, J. M.; Albani, A.: Compatibility Test for Coordination Aspects of Software Components. Australien Software Engineering Conference (ASWEC 2006). Sydney 2006; S. 41-48.
- [Zaha05] Zaha, J. M.: Automatisierte Kompatibilitätstests für fachliche Software-Komponenten. WiKu-Verlag, Berlin, 2005.
- [ZaGG04] Zaha, J. M.; Geisenberger, M.; Groth, M.: Compatibility Test and Adapter Generation for Interfaces of Software Components. In (H. Mohanty) (Hrsg.): 1st International Conference on Distributed Computing & Internet Technology (ICDCIT 2004). Bhubaneswar, India, 2004; S. 318-328.