

Martin Böttcher, Stephan Klingner (Universität Leipzig)

# Komponenten in der Dienstleistungsdomäne

---

*Ansätze zur Komponentisierung bei Dienstleistungen<sup>1</sup>*

Abstract: Die Vorteile der Komponentisierung werden bereits in den Wissenschafts- und Wirtschaftsbereichen der Softwareentwicklung und Produktion genutzt. Der vorliegende Artikel beschäftigt sich mit der Anwendung der Komponentisierung in der Domäne der Dienstleistungswirtschaft.

## 1 Einleitung

In den Wirtschafts- und Wissenschaftsbereichen der Softwareentwicklung und Produktion wird der Komponentenansatz bereits umfangreich eingesetzt, welches zu entsprechenden Vorteilen führt. Auch für die Dienstleistungsdomäne existiert die Forderung nach modular aufgebauten Dienstleistungen (Bullinger and Meiren 2001; Baida, Gordijn et al. 2003; Griebler

---

<sup>1</sup> Die in diesem Arbeitspapier spezifizierten Konzepte sind natürlichsprachlich beschrieben. Eine weitere Verwendung dieser Konzepte erfordert die formalisierte Darlegung. Dies erfolgt in einem weiteren Dokument bzw. Dokumentensammlung im Rahmen des Projekts.

2004). Dennoch existiert bislang weder ein konsistenter Ansatz zur Komponentisierung von Dienstleistungen (Akkermans, Baida et al. 2004), noch haben sich einheitliche Begriffe herausgebildet. Lediglich einzelne Dienstleistungsbranchen mit langer Entwicklungstradition wenden zunehmend den Komponentenansatz an. Hierzu zählt beispielsweise die Finanzindustrie (Baldwin and Clark 1997; Mehlaue and Wimmer 2002).

Im Folgenden sollen die Komponentenansätze im Produktions- und Softwarebereich betrachtet werden und auf die Übertragbarkeit auf die Dienstleistungsdomäne analysiert werden. Darauf aufbauend werden die Vorteile einer Komponentisierung im Dienstleistungssektor diskutiert und eine Definition des Komponentenbegriffs dargelegt. Daran anschließend werden die notwendigen Techniken zur Strukturierung von Komponenten im Sinne einer Konfigurierbarkeit spezifiziert.

## 2 Grundlagen der Komponentisierung

### 2.1 Intention der Komponentisierung

Viele Wirtschaftsbereiche sind gekennzeichnet durch zwei gegenläufige Entwicklungen:

1. Produkte werden zunehmend komplexer (Göpfert and Steinbrecher 2000).
2. Die Marktsituation erfordert die Berücksichtigung der Kundenwünsche bei gleichzeitiger Notwendigkeit der Kostenreduzierung (Peters and Sadin 2000).

Die zweite Entwicklung wird als das Problem der „Mass Customization“ (kundenindividualisierbare Massenanfertigung) bezeichnet. Hierbei wird versucht, das Spannungsfeld zwischen Standardisierung (bzw. der Massenfertigung), welche aus dem Kostendruck resultiert und der vom Markt geforderten Individualisierung aufzulösen (Gräßler 1999; Piller and Stotko 2003).

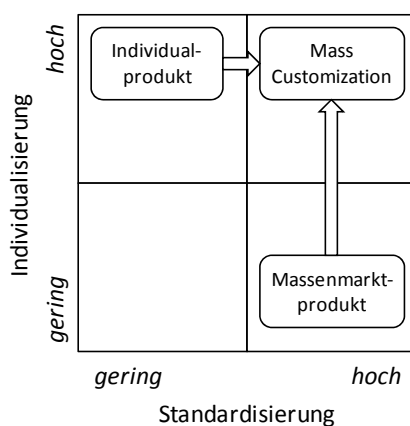


Abbildung 1: Mass Customization im Spannungsfeld zwischen Individualisierung und Standardisierung

Sowohl der zunehmenden Komplexität von Produkten als auch den Herausforderungen des Mass Customization kann durch die Modularisierung bzw. Komponentisierung begegnet werden (Pine II 1999). Durch die Bereitstellung standardisierter Komponenten und die Komposition dieser zu kundenindividuellen Gesamtprodukten können Kostenreduktion, Standardisierung und Individualisierung gleichermaßen erreicht werden (Piller and Stotko 2003). Die Modularisierung ermöglicht somit dem Kunden verschiedene Elemente so

zusammenzustellen, dass ein Produkt entsteht, das den Kundenanforderungen am besten entspricht (Baldwin and Clark 1997).

## 2.2 Komponenten im Produktionssektor

In der Produktionswirtschaft wird der Ansatz der Komponentisierung bereits seit vielen Jahren angewendet (Baldwin and Clark 1997). So definiert Starr bereits in den 60er Jahren die Notwendigkeit einer Modularisierung in der Produktion durch die Aussage: „The change that we are talking about can be briefly described as the consumer’s demand for maximum productive variety (or maximum choice). To achieve this variety, what I call „modular“ or „combinatorial“ productive capacities – that is, capacities to design and manufacture parts which can be combined in numerous ways – are required.“ (Starr 1965).

Die Notwendigkeit einer höheren Variabilität in den angebotenen Produkten und somit die Notwendigkeit nach neuen Produktions- und Angebotstechniken (bspw. durch die Anwendung von Modulen) erläutert Starr durch die drei Phasen eines Produktionszyklus, denen der Kunde folgen möchte:

1. Bei der technologischen Weiterentwicklung reagiert der Markt mit dem Wunsch, den Prototypen so umfangreich wie möglich zu verteilen. In diesem Stadium existieren nur geringe Differenzen zwischen unterschiedlichen Marken und innerhalb einer einzelnen Marke sind kaum Varianten existent. Primäres Kundenziel ist der Besitz der technischen Neuerung (bspw. MP3-Player, Handy).
2. Eine Marktveränderung setzt zu dem Zeitpunkt ein, wenn der Prototyp eine weite Verbreitung gefunden hat. Unterschiedliche Marktsegmente beginnen sich herauszubilden und die Zahl dedizierter Marken sowie die Varianten spezieller Modelle beginnen zu steigen.
3. Schlussendlich verlangt der Kunde nicht mehr nach dem reinen Besitztum des Produktes, sondern nach der Einzigartigkeit des Besitzes. Diese Veränderung entsteht aus der Tatsache, dass die Unterschiedlichkeit gegenüber anderen Kunden für den Kunden einen Mehrwert bietet (Starr 1965).

Die Ausführungen von Starr belegen deutlich die Notwendigkeit der Produktvarianten, wengleich er sich hierbei auf den B2C-Markt konzentriert.

Durch die Anwendung von Produktmodulen ist es dem Unternehmen möglich, eine wesentlich größere Variantenvielfalt zu angemessenen Kosten anzubieten (Starr 1965; Pine II 1999). Hierbei ist jedoch eine Unterscheidung zwischen den Unternehmen bzw. deren Portfolio zu treffen. Hochvolumige Industrien, die durch einen kurzen product-life-cycle gekennzeichnet sind, werden die Vorteile modularer Produkte eher ausschöpfen können, als Industrien, deren Angebot auf hochspezialisierte Einzelanfertigungen ausgelegt ist (Starr 1965). Trotz der genannten Vorteile kann die Modularisierung aber auch Probleme oder zumindest hohe Anforderungen im Erstellungsprozess mit sich bringen. Die Ersteller modularer Systeme müssen sowohl vom Gesamtprodukt als auch von den Einzelteilen (Module) Kenntnis besitzen, um die Regeln für das Zusammenspiel der Module spezifizieren

zu können. Auch treten Probleme der Modularisierung oft erst beim letztendlichen Zusammensetzen des Gesamtproduktes aus Modulen auf (Baldwin and Clark 1997).

Bezüglich der Begrifflichkeit wurde lange Zeit die Bezeichnung *Modul* geprägt. So definieren Göpfert und Steinbrecher Module als „relativ unabhängige [...] Montageeinheiten [...], die nur durch wenige aber präzise Schnittstellen miteinander verbunden sind“ (Göpfert and Steinbrecher 2000). Baldwin und Clark sprechen von einer Menge von Elementen, die abgestimmt („in concert“) zusammenarbeiten und definieren ein modulares System: „A modular system is composed of units (or modules) that are designed independently but still function as a whole“ (Baldwin and Clark 1997). Ziel eines modularen Konzepts ist es, derartige Teile zu produzieren, die eine Kombination in einer größtmöglichen Anzahl und somit unterschiedliche Produktkonfigurationen erlauben (Starr 1965). Baldwin und Clark spezifizieren die Idee der Modularisierung wie folgt: „building a complex product or process from smaller subsystems that can be designed independently yet function together as a whole“ (Baldwin and Clark 1997).

Die Intention der Modularisierung im Bereich der Produktion war die Definition einer standardisierten Menge von Schnittstellen (Interfaces) zwischen Modulen (bzw. Komponenten). Jede Komponente repräsentiert hierbei eine spezifische Funktionalität, welche über eine definierte Schnittstelle angeboten wird (Pine II 1999; Brusoni and Prencipe 2001). Dadurch kann das Design der Gesamtarchitektur vom Design der einzelnen Komponenten losgelöst werden (Brusoni and Prencipe 2001). Darüber hinaus ermöglicht die präzise Definition der Komponenten die Auslagerung der Entwicklung dieser Komponenten an Zulieferunternehmen. So werden beispielsweise die F&E-Bemühungen von Automobilunternehmen sukzessive an die Komponentenhersteller ausgelagert (Genba, Ogawa et al. 2005). Das hieraus resultierende Ergebnis ist eine Steigerung der Innovationsrate. Durch die Konzentration auf einzelne Module können sich Unternehmensbereiche bzw. Zulieferunternehmen stärker und detaillierter auf einzelne Aspekte konzentrieren (Baldwin and Clark 1997).

Ein weiterer Vorteil der Modularisierung ist die vereinfachte Handhabung der ansteigenden Komplexität von Produkten. Durch das Herunterbrechen von Produkten in Subsysteme bzw. Module gewinnen Designer, Hersteller und Kunden enorme Flexibilität (Baldwin and Clark 1997).

Entscheidend für eine gewinnbringende Modularisierung ist die Einigung auf Schnittstellen und Standards, mit denen detailliert beschrieben wird, wie Module miteinander interagieren, inklusive deren Verbindung und Kommunikation (Baldwin and Clark 1997).

### 2.3 Komponenten in der Softwareentwicklung

Die Softwareentwicklung ist gekennzeichnet durch eine Veränderung hin zur Komponentisierung. Hierbei wurden und werden unterschiedliche Entwicklungsstufen beschrrieben. Zunächst wurde das Konzept der Modularisierung umgesetzt. Hierbei stand ebenfalls das Ziel der Komplexitätsreduktion und der Verbindung der Vorteile kundenindividueller Software und standardisierter Software im Vordergrund (Szyperski 2002). Eine begriffliche Abgrenzung eines Softwaremoduls wird durch Balzert gegeben, der

für ein Modul im weiteren Sinne folgende Eigenschaften definiert: funktionale oder semantische Zusammengehörigkeit, weitgehende Kontextunabhängigkeit, Existenz definierter Schnittstellen sowie existierende Überschaubarkeit und Verständlichkeit im qualitativen und quantitativen Umfang (Balzert 2000). Eine vergleichbare Definition wird vom Duden der Informatik gegeben: „Bausteine, aus denen sich ein Softwaresystem zusammensetzt, bezeichnet man als Module. Die Beziehungen zwischen Modulen werden durch Schnittstellen festgelegt.“ (Informatik Duden).

Das Konzept der Softwaremodularisierung wurde fortgeführt und der Begriff der Softwarekomponente geprägt. Dieser Begriff wird in einem breiten Kontext genutzt, so dass eine einheitliche Definition nicht existiert. Vielmehr legen Czarnecki und Eisenecker dar, dass der Versuch einer einheitlichen Definition nicht zweckmäßig ist: „Trying to come up with a general classical definition for software components is not only futile, but harmful“ (Czarnecki and Eisenecker 2000). Nichtsdestotrotz werden Definitionen und Eigenschaften von Softwarekomponenten in der wissenschaftlichen Literatur diskutiert.

Die Intention komponentenbasierter Softwareentwicklung kann in Analogie zu den vorgefertigten Bauteilen in der Automobilindustrie gesehen werden. Das Ziel besteht in der Kombination von bestehenden Komponenten zu einem neuen Produkt (Frank 1999; Pfeiffer and Winkelmann 2007). Auch soll die Duplikation von Code reduziert und die Wiederverwendung maximiert werden (Czarnecki and Eisenecker 2000).

Die Vorteile komponentenbasierter Softwareentwicklung sind vielfältiger Natur. So soll die Qualität von Software bei gleichzeitig deutlich geringeren Kosten nachhaltig verbessert werden (Frank 1999). Auch soll die Produktivität bei der Gestaltung von Anwendungssystemen durch die Wiederverwendung gesteigert werden, indem erhebliche Zeit- und Kostenvorteile erzielt werden (Sametinger 1997; Gao, Ma et al. 2006). Neben der einfacheren, schnelleren und preiswerteren Herstellung von Anwendungen mit Hilfe von vorgefertigten Komponenten werden auch Verbesserungen bei der Wartbarkeit von Anwendungen erreicht (Balzert 2000).

In der wissenschaftlichen Literatur wird eine Vielzahl konkreter Definitionen des Komponentenbegriffs in der Softwareentwicklung gegeben, ohne explizit den Anspruch zu erheben, allgemeingültig zu sein:

- „A software component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be developed independently and is subject to composition by third parties.“ (Szyperski 2002)
- „Components are artefacts that we clearly identify in our software systems. They have an interface, encapsulate internal details and are documented separately.“ (Sametinger 1997)
- „Unter einer Komponente kann ein unabhängiger, wieder verwendbarer Softwarebaustein verstanden werden, der eine Schnittstellenbeschreibung sowie eine Spezifikation seiner funktionalen und nicht-funktionalen Eigenschaften umfasst.“ (Pfeiffer and Winkelmann 2007)

- „We define software components simply as building blocks from which different software systems can be composed. [...] we want them to be plug-compatible by design and to be combinable in as many ways as possible.“ (Czarnecki and Eisenecker 2000)
- „Ein Halbfabrikat bzw. eine Komponente (component) ist ein abgeschlossener, binärer Software-Baustein, der eine anwendungsorientierte, semantisch zusammengehörende Funktionalität besitzt, die nach außen über Schnittstellen zur Verfügung gestellt wird. Beim Entwurf des Halbfabrikats wurde auf hohe Wiederverwendung Wert gelegt.“ (Balzert 2000)

Zu den typischen Eigenschaften einer Softwarekomponente gehören unter anderem:

- Plattformunabhängigkeit,
- Wiederverwendbarkeit,
- Zugang über eine Schnittstelle sowie
- Kapselung der Funktionalität (Sametinger 1997; Frank 1999).

Eine aktuelle Weiterentwicklung des Modul- und Komponentenkonzeptes repräsentiert der Ansatz der serviceorientierten Architekturen. In diesem Fachgebiet werden sogenannte Softwareservices definiert: „A service is generally implemented as a coarse-grained [sic – coarse-grained], discoverable software entity that exists as a single instance with applications and other services through a loosely coupled (often asynchronous), message-based communication model“ (Brown, Johnston et al. 2002).

### 3 Komponenten der Dienstleistungsdomäne

#### 3.1 Zielstellung der Komponentisierung von Dienstleistungen

Wie auch bei der Komponentisierung von Produkten und Software ergibt sich die Notwendigkeit der Komponentisierung von Dienstleistungen aus Veränderungen in der Dienstleistungswirtschaft. Anstelle einzelner abgegrenzter Dienstleistungen werden zunehmend komplexe, variantenreiche, industrielle Dienstleistungen am Markt angeboten, so dass Dienstleistungen konfiguriert und gebündelt werden müssen (Schramm and Pallentien 2004). Wenngleich die Besonderheiten der Dienstleistungsdomäne dazu führten, dass der Fokus wissenschaftlicher Arbeiten oft auf Aspekte der Kundeninteraktion und anderer weicher Faktoren gelegt wurde, stehen Dienstleistungsanbieter aufgrund der Veränderung des Dienstleistungssektors zunehmend vor der Herausforderung, bestimmte Bereiche ihrer Dienstleistungsangebote unter wirtschaftlich-pragmatischen Gesichtspunkten vergleichbar zum Produktionssektor zu betrachten. Insbesondere umfangreiche Dienstleistungsportfolios sowie die Notwendigkeit der Standardisierung bei gleichzeitiger Kundenindividualität erfordern Ansätze wie bspw. die Komponentisierung von Dienstleistungen (Baida, Gordijn et al. 2003). Die Auseinandersetzung mit dem Modularisierungspotential bei Dienstleistungen wird in der wissenschaftlichen Literatur kaum betrieben (Burr 2002).

Prinzipiell liegen die Vorteile modularer Servicearchitekturen in einer „effizienteren Exploration der bestehenden unternehmerischen Ressourcenpotentiale und/oder in einer

rascheren und effizienteren Exploration neuer Ressourcenkombinationen“ (Burr 2002). Die einzelnen mit einem komponentenorientierten Ansatz verfolgten Ziele lassen sich wie folgt darstellen:

- **Aufwandsreduktion:** Der hohe Aufwand bei der Preisgestaltung, der Ressourcenallokation und dem Management monolithischer Dienstleistungen kann durch einen komponentenbasierten Ansatz reduziert werden.
- **Konfiguration:** Mithilfe von Dienstleistungskomponenten soll die Möglichkeit der strukturierten Konfiguration kundenindividueller Dienstleistungen gegeben werden. Ein Portfolio standardisierter Komponenten aus dem Kunden eine Gesamtleistung konfigurieren können, wird bereitgestellt. Somit wird eine Balance zwischen Standardisierung und individuellem Angebot geschaffen.
- **Weiterentwicklung und Verbesserung:** Einzelne abgegrenzte Komponenten lassen sich besser weiterentwickeln, da deren Interdependenz zum Gesamtangebot über wohldefinierte Schnittstellen definiert wird. Die Abhängigkeiten innerhalb eines monolithischen Angebots lassen sich wesentlich schwerer überblicken, so dass singuläre Verbesserungen nur schwer umgesetzt werden können.
- **Wiederverwendung:** Planung, Umsetzung und Verbesserung von Dienstleistungsangeboten erfolgen mit einem entsprechend hohen Aufwand. Um Skaleneffekte zu nutzen und den Aufwand zu relativieren, können Dienstleistungskomponenten genutzt werden, um Angebote wiederzuverwenden. Einmal entwickelte Dienstleistungskomponenten können in unterschiedlichen Gesamtangeboten erneut ihre Verwendung finden.
- **Übersichtlichkeit und Komplexitätsreduktion:** Dienstleistungsunternehmen, die regelmäßig neue Dienstleistungen entwickeln, besitzen ein umfangreiches Dienstleistungsportfolio. Die Strukturierung des Portfolios durch die Anwendung von Dienstleistungskomponenten führt zu einer größeren Übersichtlichkeit. Auch die Weiterverarbeitung durch elektronische Kataloge oder Konfiguratoren wird durch Dienstleistungskomponenten unterstützt.

## 3.2 Definition des Komponentenbegriffs für die Dienstleistungsdomäne

### 3.2.1 Definitionsansätze

Die erwarteten Vorteile der Komponentisierung von Dienstleistungen führen zu einer zunehmenden Auseinandersetzung mit dieser Thematik in der wissenschaftlichen Literatur. Wenngleich die verfolgte Zielrichtung zwischen den verschiedenen Ausführungen vergleichbar ist, so lässt sich insbesondere bei der verwendeten Terminologie eine hohe Heterogenität feststellen. So werden die Begriffe „Dienstleistungskomponente“ (Corsten, Dresch et al. 2006; Burianek, Ihl et al. 2007; Thomas, Walter et al. 2008), „Dienstleistungsmodul“ (Meier, Maßberg et al. 2003; Bäcker and Herzog 2004; Schramm and Pallentien 2004; Corsten, Dresch et al. 2006; Herrmann, Klein et al. 2006) und „Dienstleistungsbaustein“ (Herrmann and Klein 2004; Emmrich 2005; Thomas and Scheer 2006) synonym verwendet. Wenngleich diese Begriffe verwendet werden, so sind sie dennoch

nur selten exakt definiert und erklären sich somit eher aus dem Kontext des Textes heraus. Nachfolgend werden auszugsweise einige Definitionen wiedergegeben, welche unter anderem auch die Synonymität der Begriffe aufzeigen:

<i>Definition</i>	<i>Quelle</i>
„Ein modulares Dienstleistungssystem umfasst mehrere definierte Teilprozessblöcke (Module), die zur Befriedigung heterogener Kundenbedürfnisse ein Spektrum an Kombinationsmöglichkeiten, die keine Systemveränderung erfordern, bieten.“	(Corsten, Dresch et al. 2007)
„Ein Dienstleistungsbaustein ist dabei eine geschlossene logische Gesamtheit, die eine betriebswirtschaftlich sinnvolle und eindeutig abgegrenzte Komponente einer Dienstleistung darstellt. Die Assoziationen zwischen den Dienstleistungsbausteinen konkretisieren sich in aufbau- (Produktmodell) und ablauflogischen (Prozessmodell) Anordnungsabhängigkeiten. Dienstleistungsbausteine können auf mehreren Ebenen in eigenständige Dienstleistungsbausteine dekomponiert werden. Sie stellen gewissermaßen die Produktbaugruppen der Dienstleistungen dar.“	(Thomas and Scheer 2006)
„Modulare Servicearchitekturen beruhen auf dem Prinzip der Dekomposition. Eine komplexe Dienstleistung (ein Kundenproblem) wird in eine strukturierte Anordnung von Dienstleistungsmodulen (Teilproblemen) zerlegt. [...] Ein Dienstleistungsmodul wird somit verstanden als Einheit von Teilfunktion, Teildienstleistung und ausführendem Aufgabenträger. Durch Definition von standardisierten Schnittstellen, die Ziele, Regeln und Verhaltensspielräume für die Interaktion zwischen den Dienstleistungsmodulen vorgeben, wird ein System von lose gekoppelten Dienstleistungsmodulen generiert.“	(Burr 2002)

Aus den obigen Definitionsansätzen und den Ausführungen der Komponentenbetrachtung anderer Disziplinen lässt sich folgende Definition einer Dienstleistungskomponente ableiten:

*Eine Dienstleistungskomponente ist eine definierbare Funktionalität, welche eine abgegrenzte Menge von Dienstleistungssystemelementen von einem Zustand in einen anderen Zustand überführt.*

Eine Dienstleistungskomponente kann somit als ein abgegrenztes Dienstleistungssystem verstanden werden (Böttcher and Fähnrich 2009). Zu den Systemelementen gehören Anbieter, Nachfrager sowie deren Ressourcen<sup>2</sup>. Die definierbare Funktionalität einer Dienstleistungskomponente repräsentiert die angebotene Dienstleistung und bietet einen Mehrwert für den Nachfrager.

Die Idee der Dienstleistungskomponentisierung beinhaltet ebenfalls, dass sich Komponenten aus anderen Komponenten zusammensetzen können.<sup>3</sup> Diese Zusammensetzung ist somit derart gestaltet, dass eine Komponente letztendlich die Summe aus anderen Komponenten

<sup>2</sup> Im engeren Sinne handelt es sich auch beim Anbieter und Nachfrager selbst um Ressourcen.

<sup>3</sup> Dieser Ansatz wird auch in der Produktionstechnik verwendet. Dort setzt sich bspw. die Komponente Motor ihrerseits wiederum aus Schrauben, Kolben, etc. zusammen.



darstellt. Somit muss eine Komponente immer aus mindestens zwei weiteren Komponenten zusammengesetzt werden. Komponenten, bei denen sich eine Komponente aus exakt einer anderen Komponente zusammensetzt, sind äquivalent und somit redundant. Diese Tatsache ist bei der Erstellung von Konfigurationsgraphen (siehe Abschnitt 5) zu berücksichtigen.

### 3.2.2 Anforderungen an Dienstleistungskomponenten

In der wissenschaftlichen Literatur existieren Darstellungen zu den Eigenschaften von Dienstleistungskomponenten bspw. (Böhmman, Junginger et al. 2002), (Burr 2002), (Meier, Maßberg et al. 2003), (Corsten, Dresch et al. 2007). Diese können um die Erkenntnisse aus den oben gegebenen Dienstleistungsdefinitionsansätzen erweitert werden. Daraus ergeben sich folgende Anforderungen an Dienstleistungskomponenten:

- Geringe Kopplung zwischen Komponenten,
- Abgeschlossenheit einer Komponente,
- Möglichkeit der Dekomposition und Komposition sowie
- Wohldefinierte Schnittstellen.

Aus diesen Anforderungen ergibt sich, dass Dienstleistungskomponenten zu individuellen Gesamtlösungen kombiniert und Komponenten für verschiedene Gesamtlösungen wiederverwendet werden können. Darüber hinaus lassen sich logische Anforderungen ableiten wie bspw. die präzise Dokumentation, welche die Grundlage der Schnittstellendefinition und der Kombinierbarkeit bildet.

### 3.3 Bildung von Dienstleistungskomponenten

Die Bildung von Dienstleistungskomponenten ist immer in Zusammenhang mit dem damit verfolgten Kontext zu betrachten. Ein allgemeingültiges Vorgehen oder Muster existiert nicht. Beispielhaft seien die Arbeiten von Burr (Burr 2002), Hermsen (Hermsen 2000) und Böhmman et al. (Böhmman, Junginger et al. 2002) für die Erstellung von Dienstleistungskomponenten aufgeführt. Zugleich können aber auch Arbeiten aus der Produktionsforschung verwendet werden, bspw. (Piller and Stotko 2003) und (Peters and Sadin 2000).

Die genaue Erstellung einzelner Dienstleistungskomponenten muss somit den Eigenschaften einer Dienstleistungskomponente gerecht werden. Darüber hinaus ist eine Abwägung zwischen dem Aufwand und dem Nutzen der Komponentisierung von Dienstleistungsangeboten zu berücksichtigen. Insbesondere sich oft wiederholende Abläufe, die sich gut strukturieren und standardisieren lassen, eignen sich hierfür. Dagegen muss die Eignung der Komponentenbildung für Unternehmen mit kundenindividuellen Einzelangeboten genau geprüft werden.

Die Komponentisierung benötigt Wissen über

1. „die von der gesamten Servicearchitektur und einzelnen Dienstleistungsmodulen zu erfüllenden Funktionen

2. die Zuordnung einzelner Dienstleistungsfunktionen zu einzelnen Dienstleistungsmodulen
3. die Interaktionen verschiedener Dienstleistungsmodule innerhalb der gesamten Servicearchitektur und die Definition von Schnittstellen zwischen den Dienstleistungsmodulen
4. die Zuordnung von Dienstleistungsmodulen zu unternehmensinternen oder unternehmensexternen Modullieferanten
5. die Koordination der Modullieferanten und die Integration der Dienstleistungsmodule zu umfassenden Kundenlösungen.“ (Burr 2002)

#### **4 Eigenschaften von Dienstleistungskomponenten**

Mit der Spezifikation einzelner Dienstleistungskomponenten ergibt sich die Notwendigkeit, diese präzise beschreiben zu können, um deren Funktionalität sowie weitere Eigenschaften (bspw. Preis, Ort, etc.) darzulegen. Aufbauend auf dieser Beschreibung können Komponenten identifiziert und Angebote für Kunden erstellt werden.

Prinzipiell lassen sich zwei Eigenschaftsklassen definieren:

1. Funktionale Eigenschaften: Mit diesen Eigenschaften wird die von einer Komponente angebotene Funktionalität beschrieben.
2. Nicht-funktionale Eigenschaften: Diese Eigenschaften definieren Restriktionen auf die funktionalen Eigenschaften und grenzen die Funktionalität ein. Zu diesen Eigenschaften gehören bspw. der Erbringungsort, der Erbringungszeitraum aber auch der Preis und Bezahloptionen.

Im Nachfolgenden werden die beiden Eigenschaftsklassen in Ansätzen dargestellt. Für eine präzise Darlegung einzelner Eigenschaftsattribute sowie die Erstellung darauf aufbauender Modelle sei bspw. auf die Arbeiten (Böttcher 2009) und (O'Sullivan 2006) verwiesen.

##### **4.1 Funktionale Eigenschaften**

Funktionale Eigenschaften erlauben die Beschreibung der angebotenen Funktionalität einer Komponente. Sie spezifizieren somit die von der Komponente vorgenommenen Zustandsänderungen. Die funktionalen Eigenschaften lassen sich textuell nichtformal und formal darstellen. Die textuelle Beschreibung ermöglicht die Repräsentation der Eigenschaft in einer für den Menschen verständlichen Art. Sie findet beispielsweise in der Erstellung von Broschüren und Prospekten ihre Anwendung. Darüber hinaus können die Eigenschaften aber auch formal beschrieben werden, wodurch die Weiterverwendbarkeit durch IT-Systeme wesentlich verbessert wird. Hierzu zählt beispielsweise aus dem Forschungsbereich der künstlichen Intelligenz die Planung, welche Aktionen definiert, die die Veränderung von Zuständen zur Folge hat (Russel and Norvig 2004).

Die funktionalen Eigenschaften lassen sich neben der expliziten textuellen oder formalen Darlegung auch implizit spezifizieren. Hierfür dienen:

- **Ressourcenorientierte Funktionalität.** Dienstleistungskomponenten dienen immer der Veränderung eingebrachter Ressourcen. Die Zustandsänderung der Ressourcen über den Zeitverlauf der Dienstleistungserbringung ermöglicht die implizite Darlegung angebotener Funktionalität.

Hierbei lässt sich die Veränderung einer Ressource entsprechend der Veränderung seiner Attribute (bspw. Änderung der Farbe) oder entsprechend der Veränderung der Beziehung zu anderen Ressourcen (bspw. von „lose gekoppelt“ zu „verschweißt“) darstellen.

- **Klassifikation und Typisierung.** Durch die Klassifikation bzw. die Typisierung von Dienstleistungen werden gleichartige Dienstleistungen zusammengefasst und einer Klasse zugeordnet. Die Zusammenfassung erfolgt hierbei meist aufgrund der Ähnlichkeit der angebotenen Funktionalität (bspw. „Reinigungsleistung“). Die Klassifikation von Dienstleistungen kann somit ebenfalls genutzt werden, um die Funktionalität einer Dienstleistung implizit wiederzugeben. Für die Klassifikation stehen unterschiedliche Standards zur Verfügung. Hierzu zählen bspw. eCl@ss, ETIM und UN/SPSC.
- **Ziele.** Ziele repräsentieren die verfolgte Zustandsänderung durch die angebotene Dienstleistung. Prinzipiell kann zwischen internen (Ziele, die durch das anbietende Unternehmen verfolgt werden) und externen Zielen (Ziele, die der Kunde verfolgt) unterschieden werden. Meist befinden sich diese Ziele in einem gegenläufigen Spannungsverhältnis, so dass es die Aufgabe des anbietenden Unternehmens ist, eine Abwägung zwischen diesen beiden Zielen zu finden.

Für die Wiedergabe der verfolgten Ziele lassen sich diese entweder textuell oder durch die Zuhilfenahme von Skalen spezifizieren. Für die Anwendung der Skalen existieren verschiedene Formen mit unterschiedlichen Aussagegegenständen: Nominalskala, Ordinalskala, Intervallskala und Verhältnisskala (Teschl and Teschl 2006).

#### 4.2 Nicht-funktionale Eigenschaften

Während die funktionalen Eigenschaften die angebotene Funktionalität einer Dienstleistungskomponente beschreiben, dienen die nicht-funktionalen Eigenschaften der Charakterisierung der Randbedingungen offerierter Funktionalitäten. Nicht-funktionale Eigenschaften sind somit jene Attribute einer Dienstleistungskomponente, die als Restriktionen oder Grenzen auf die Funktionalität von Dienstleistungen einwirken, ohne selbst Funktionalität zu repräsentieren.

Zu den nicht-funktionalen Eigenschaften gehören:

- **Identifizierung:** Diese nicht-funktionale Eigenschaft dient der eindeutigen Wiedererkennung einer Dienstleistungskomponente und hat somit rein administrativen Charakter.
- **Anbieter:** Dieses Attribut identifiziert die offerierte Geschäftseinheit, welche zugleich die Verantwortung für die Dienstleistungskomponente trägt. Für die nähere Spezifikation können Ansätze wie White Pages (Bettag 2001) oder Standards wie UDDI

(<http://uddi.xml.org/>) herangezogen werden. So gehören bspw. der Bezeichner und Kontaktinformationen zur näheren Spezifikation des Anbieters.

- **Zeitliche Verfügbarkeit:** Hierdurch wird dargelegt, wann eine Dienstleistung verfügbar ist. Im Speziellen ist festzulegen, zu welchem Zeitpunkt eine Ressource in das Dienstleistungssystem zu integrieren ist und wann eine Ressource als Ergebnis einer Dienstleistungserbringung entnommen werden kann. Für die Wiedergabe der zeitlichen Verfügbarkeit lassen sich Zeitpunkte, Zeiträume sowie Ausnahmen definieren.

Für die Spezifikation der Zeitpunkte werden das Datum und die Uhrzeit angegeben. Das Datum kann durch kalendarische Daten, Ordinaldaten und Wochendaten wiedergegeben werden (ISO 2004). Die Zeit wird durch die Angabe der Stunde, Minute und Sekunde und die Spezifikation der Zeitzone definiert. Neben einzelnen Zeitpunkten lassen sich auch wiederkehrende Zeitpunkte (bspw. „jeden Freitag“) festlegen, indem die Datumsangaben mit einer höheren Granularität gewählt werden.

Neben der Definition einzelner Zeitpunkte lassen sich auch Zeiträume abgrenzen. Dies erfolgt durch die Angabe zweier Zeitpunkte, die den Anfangs- und Endzeitpunkt markieren.

Um die Angabe von Zeiträumen oder wiederkehrenden Zeitpunkten einschränken zu können, werden Ausnahmen definiert, die ihrerseits Zeitpunkte oder Zeiträume sind. So lassen sich bspw. Aussagen wie „jeden Freitag außer der erste Freitag im Mai“ repräsentieren.

- **Lokationsverfügbarkeit:** Hierdurch wird festgelegt, an welchem Punkt eine Dienstleistung zur Verfügung steht bzw. präziser definiert, an welchem Punkt Ressourcen in das Dienstleistungssystem einzubringen oder zu entnehmen sind. Hierbei wird der Begriff des „Punktes“ so weit gefasst, dass neben geographischen Angaben (bspw. Adressen oder Längen und Breitengrade) auch Telefonnummern, Internetadressen oder Spektren von elektromagnetischen Wellen definiert werden können. Vergleichbar zur zeitlichen Verfügbarkeit lässt sich die Lokationsverfügbarkeit als einzelner Punkt oder Wertebereich zwischen zwei Punkten definieren. Hier können wiederum Ausnahmen beschrieben werden, welche Aussagen erlauben wie bspw. „Hausnummer 1-100 außer 13-15“.
- **Zeitliche Dauer:** Die zeitliche Dauer legt den Zeitraum zwischen der Integration einer Kundenressource in das Dienstleistungssystem und dem Entnehmen der Ressource aus dem Dienstleistungssystem fest und markiert somit den zeitlichen Aufwand für die Erbringung einer Dienstleistung. Die zeitliche Dauer ist hierbei von anderen nicht-funktionalen Eigenschaften (bspw. zeitliche Verfügbarkeit und Lokationsverfügbarkeit) abhängig. Darüber hinaus kann eine Varianz einer zeitlichen Verfügbarkeit definiert werden bspw. „zwischen 13 und 15 Stunden“.
- **Preis:** Der Preis definiert den Gegenwert, der für die Erbringung einer Dienstleistung seitens des Leistungsempfängers zu erbringen / zahlen ist. Die Preisdefinition obliegt hierbei einer hohen Komplexität (Leukel and Schmitz 2002). Prinzipiell müssen einzelne

Preiskomponenten festgelegt werden bspw. „einmalige Zahlung oder monatliche Zahlungen“. Diese lassen sich als Absolutpreis, anteiliger Preis, Preisfunktion oder dynamischer Preis angeben (Leukel and Schmitz 2002). Darüber hinaus müssen Angaben zur Währung sowie der im Preis enthaltenen oder nicht enthaltenen Steuern gegeben werden. Des Weiteren ist der Preis in Abhängigkeit zu anderen nicht-funktionalen Eigenschaften zu setzen. So kann dieser bspw. je nach Ort oder Zeit variieren. Letztendlich können noch Festlegungen bzgl. der Verhandelbarkeit des Preises gemacht werden.

- **Bezahlung:** Hierbei wird definiert, wie und unter welchen Restriktionen ein Kunde den angegebenen Preis dem Anbieter zu überlassen hat. Hierbei sind einerseits das Instrument (bspw. Kreditkarte) sowie damit verbundene Restriktionen (bspw. nur in bestimmten Regionen) zu spezifizieren.
- **Konsequenzen:** Die Konsequenzen umfassen alle Auswirkungen, die sich aus einer Nichteinhaltung von Vereinbarungen ergeben. Diese treffen sowohl für den Anbieter als auch für den Kunden zu. Konsequenzen dienen meist als Ausgangslage für die Definition von Service-Level-Agreements oder vergleichbaren Vertragswerken.

## 5 Konfiguration von Dienstleistungskomponenten

Ein wesentliches Ziel der Komponentisierung ist die Zusammenstellung eines Gesamtangebots aus einzelnen Dienstleistungskomponenten. Dieser Vorgang wird unter dem Begriff der Konfiguration zusammengefasst. Aus standardisierten Komponenten soll eine kundenindividuelle Gesamtlösung zusammengesetzt werden, wobei Abhängigkeiten zwischen den einzelnen Komponenten berücksichtigt werden müssen.

### 5.1 Grundlagen der Konfiguration

Prinzipiell lässt sich der Begriff der Konfiguration unterscheiden in einen Zustand oder eine Aktivität. Stellt die Konfiguration einen Zustand dar, dann handelt es sich um eine konkrete Zusammenstellung aus einer Menge von Geräten oder Bauteilen. Demgegenüber definiert die Konfiguration als Aktivität den Prozess der Auswahl aus einer bestimmten Anzahl von vorgegebenen Geräten oder Bauteilen bzw. die Adaption vorgegebener Konstellationen. Diese Anpassung erfolgt hierbei immer unter Berücksichtigung vorgegebener Regeln und Abhängigkeiten (Gabler 2000; o.V. 2001). Im Rahmen dieses Abschnitts wird bei der Konfiguration immer von der Aktivität des Konfigurierens gesprochen.

Das mit der Konfiguration zusammenhängende Konfigurationsproblem lässt sich formal spezifizieren (Stormer 2007):

Ein Konfigurationsproblem

$$C$$

ist ein Tripel

$$C=(X,D,F)$$

mit

Komponenten in der Dienstleistungsdomäne

$$X = \{x_1, \dots, x_n\}$$

der Menge der Optionen. Für jede Option  $x_i$  aus der Menge  $X$  wird eine endliche Menge von Optionswerten  $D_i$  definiert. Die Menge  $D$  ist das kartesische Produkt all dieser Optionswerte

$$D = D_1 \times D_2 \times \dots \times D_n$$

In der Menge

$$F = \{f_1, f_2, \dots, f_3\}$$

werden die einschränkenden Funktionen abgelegt. Eine gültige Konfiguration wählt für jede Option  $x_i$  aus der Menge  $X$  genau einen Optionswert

$$d_j \in D_i,$$

so dass alle Regeln aus  $F$  erfüllt sind.

Zusammenfassend lässt sich feststellen, dass für die Darlegung der Konfiguration Optionen mit ihren möglichen Werten sowie einschränkende Regeln spezifiziert werden müssen.

## 5.2 Konfigurationsansatz für die Dienstleistungsdomäne

Insbesondere in der Produktion findet die Konfiguration bereits verbreitete Anwendung (Dabic 2005). Hierfür werden Stücklisten unterschiedlicher Form und Aussagekraft erstellt. Prinzipiell lässt sich mit Stücklisten darlegen, dass Sachgüter aus Teilen zusammengesetzt werden (Becker and Overhage 2003). Zu den gängigsten Formen von Stücklisten gehören Mengenübersichtsstückliste, Strukturstückliste, Baukastenstückliste, Variantenstückliste, Gleichteilestückliste und Plus-Minus-Stückliste (Blohm, Beer et al. 1997).

Die sich aus der Konfiguration von Sachgütern ergebenden Vorteile sollen auch auf die Domäne der Dienstleistung übertragen werden. Die Spezifizierung von Dienstleistungen erfolgt hierbei in zwei Schritten. Einerseits müssen Komponenten zu Gesamtangeboten zusammengefügt werden und andererseits müssen die Eigenschaften der einzelnen Komponenten angepasst werden.

### 5.2.1 Konfigurationsgraph

Grundlegend soll mit dem Konfigurationsgraph dargelegt werden, dass sich eine Dienstleistungskomponente aus anderen Dienstleistungskomponenten zusammensetzen kann (Böhmman, Junginger et al. 2002). Die hierfür benötigte Beziehung zwischen den einzelnen Komponenten wird durch Begriffe wie „hat Teil“, „besteht aus“ (Grieble 2004), „ist übergeordnet zu“ (Herrmann and Klein 2004) oder „has-parts“ (Cunis and Günter 1991) beschrieben. Darüber hinaus müssen sich Variationen der Zusammensetzung spezifizieren lassen, um die Konfiguration durch einen Kunden zu ermöglichen.

Für ein besseres Verständnis der Zusammensetzung einzelner Komponenten bedarf es einer Repräsentation der Struktur. Darüber hinaus soll diese redundanzfrei erfolgen, um den Pflegeaufwand zu reduzieren und die Gefahr von Inkonsistenzen zu minimieren. Eine redundanzfreie Repräsentation derartiger Strukturen wird in der Produktmodellierung durch

sogenannte Gozinto-Graphen<sup>4</sup> erreicht. Durch diesen Graphen werden die Teilerzeugnisse und Enderzeugnisse nur einmal wiedergegeben (Zahn and Schmidt 1996). Mit dem Gozinto-Graphen in seiner ursprünglichen Form können zwar die Strukturen gut abgebildet, jedoch keine Varianten definiert werden (Becker and Overhage 2003).

Um die Vorteile der strukturellen Repräsentation durch einen Gozinto-Graphen nutzen zu können und zugleich die Konfiguration von Dienstleistungen zu ermöglichen, muss der Ansatz des Konfigurationsgraphen erweitert werden. Zunächst wird ein Netzgraph definiert, der die Auswahl zwischen unterschiedlichen Komponenten ermöglicht.

Ein Netzgraph zur Konfiguration von Dienstleistungen besteht zunächst aus Knoten, welche die einzelnen Dienstleistungskomponenten repräsentieren. Diese Knoten können durch die Namen der Dienstleistungskomponenten identifiziert werden. Mit den gerichteten Verbindungslinien wird spezifiziert, dass Dienstleistungskomponenten in andere Dienstleistungskomponenten eingehen bzw. sich Dienstleistungskomponenten aus anderen zusammensetzen (Abbildung 2).

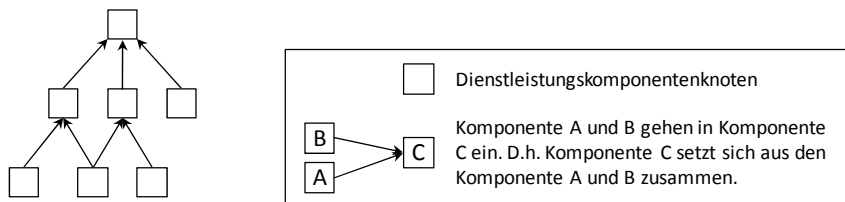


Abbildung 2: Graph mit Dienstleistungskomponenten und gerichteten Verbindungskanten

Mit diesem Graph lassen sich Gozinto-Graphen abbilden. Hierbei ist jedoch die geforderte Auswahl zwischen unterschiedlichen Teilkomponenten nicht möglich. Um dieses ausdrücken zu können, bedarf es der Definition zusätzlicher Semantik. Diese kann entweder für die jeweiligen Kanten oder durch zusätzliche Definition neuer Knotentypen spezifiziert werden. Für eine bessere Übersichtlichkeit werden für den vorgestellten Graphen zusätzliche Knoten festgelegt, die konfigurationsrelevante Semantik besitzen. Demzufolge wird ein neuer Knotentyp (Verbindungsknoten) definiert, der die Verbindung zwischen den einzelnen Dienstleistungskomponenten regelt (Abbildung 3).

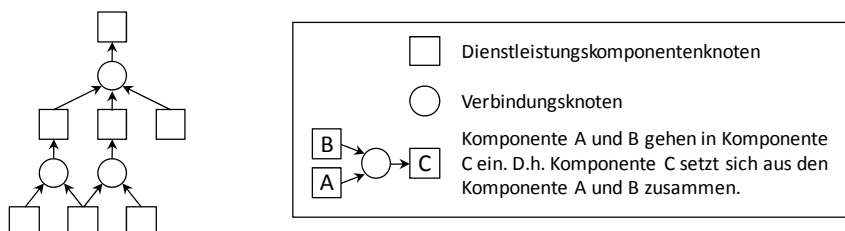


Abbildung 3: Graph mit Dienstleistungsknoten, Verbindungsknoten und gerichteten Verbindungskanten

Die Verbindungsknoten dienen als Ausgangspunkt für die Konfiguration, indem diesen Knoten eine unterschiedliche Semantik (bspw. „und“ / „oder“ (Becker and Overhage 2003))

<sup>4</sup> Gozinto-Graphen wurden erstmals von Vazonyi als sogenannte Gozinto-Diagramme definiert Vazonyi, A. (1962). Die Planungsrechnung in Wirtschaft und Industrie. Wien et al., Oldenbourg.

zugeordnet wird. Hierfür werden fünf verschiedene Verbindungsknoten spezifiziert (Abbildung 4).

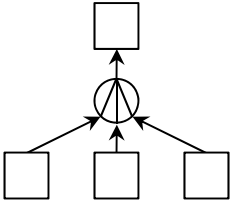
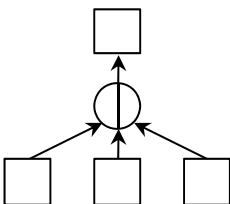
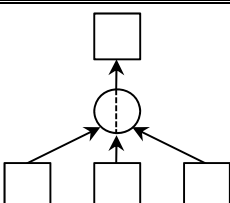
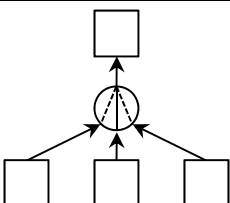
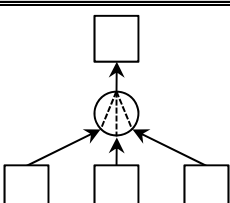
Konjunktivknoten	
Dieser Knoten definiert, dass sich eine Komponente aus <b>allen</b> über diesen Knoten verbundenen Komponenten zusammensetzen muss.	
Exklusives-Oder-Knoten	
Dieser Knoten definiert, dass sich eine Komponente aus <b>genau einer</b> der über diesen Knoten verbundenen Komponenten zusammensetzen muss. Diese Verbindung wird insbesondere für die Kombination von Verbindungsknoten benötigt (s.u.).	
Optional-Exklusives-Oder-Knoten	
Dieser Knoten definiert, dass sich eine Komponente aus <b>genau einer oder keiner</b> der über diesen Knoten verbundenen Komponenten zusammensetzen muss. Dieser Verbindungsknoten wird insbesondere für die Kombination von Verbindungsknoten benötigt (s.u.).	
Disjunktiv-Obligatorischer-Knoten	
Dieser Knoten definiert, dass sich eine Komponente aus <b>mindestens einer</b> der über diesen Knoten verbundenen Komponenten zusammensetzen muss. Dieser Verbindungsknoten wird insbesondere für die Kombination von Verbindungsknoten benötigt (s.u.).	
Disjunktiv-Knoten	
Dieser Knoten definiert, dass sich eine Komponente aus <b>keiner oder beliebig vielen</b> der über diesen Knoten verbundenen Komponenten zusammensetzen muss. Dieser Verbindungsknoten wird insbesondere für die Kombination von Verbindungsknoten benötigt (s.u.).	

Abbildung 4: Verbindungsknotentypen

Neben der direkten Verbindung von Dienstleistungskomponenten können die Verbindungsknoten auch genutzt werden, um ihrerseits Verbindungsknoten logisch miteinander zu verbinden. Das Beispiel (Abbildung 5) zeigt einen Verbindungsknoten, der



darstellt, dass einer der beiden nachfolgenden Verbindungsknoten gewählt werden muss. Die nachfolgenden Verbindungsknoten besitzen ihrerseits wiederum Semantik zur Aussage, welche Dienstleistungskomponenten im Rahmen einer Konfiguration zu wählen sind.

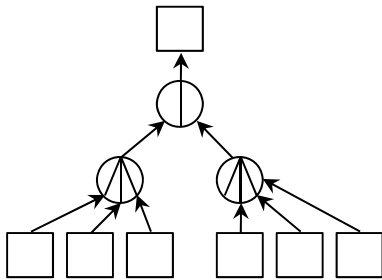


Abbildung 5: Beispielhafte Verknüpfung von Verbindungsknoten

Aus der Definition der Komponenten lässt sich ableiten, dass die Erstellung eines Konfigurationsgraphen und der Einsatz der Verbindungsknoten immer derart zu gestalten ist, dass eine Komponente immer aus mehreren Komponenten zusammengesetzt wird. Anderenfalls muss eine Veränderung des Graphen vorgenommen werden (siehe Abbildung 6).

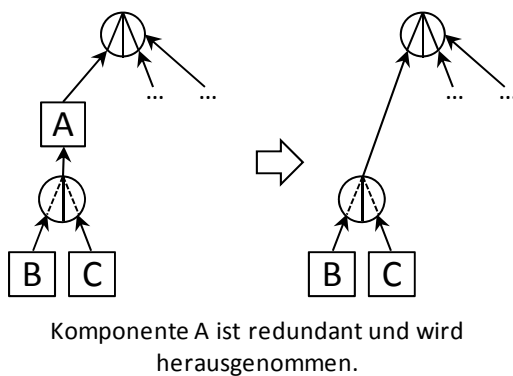


Abbildung 6: Beispiel eines inkonsistenten Graphen

Mit der Menge der dargelegten Verbindungsknoten lässt sich eine Vielzahl von Konfigurationsmöglichkeiten darlegen. Beispielhaft wird das in Abbildung 7 dargelegt.

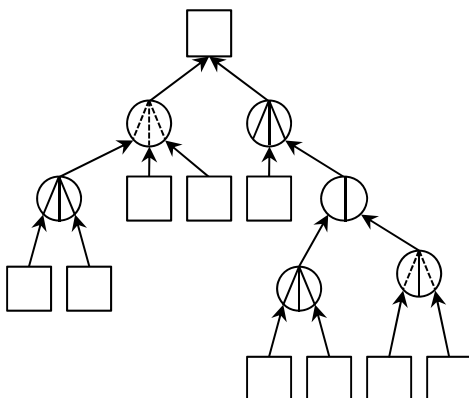


Abbildung 7: Beispielhafter Konfigurationsgraph

Wenngleich die Modellierung oben spezifizierter Graphen relativ einfach ist, ist zugleich jedoch auch dessen Ausdrucksmächtigkeit eingeschränkt. So lassen sich bspw. keine Komponenten in der Dienstleistungsdomäne

Aussagen wie „Eine Komponente setzt sich aus 4-6 der nachfolgenden Komponenten zusammen.“ treffen. Hierfür bedarf es einer Erweiterung des Graphs unter Zuhilfenahme von Kardinalitäten.

Derartige Kardinalitäten werden beispielsweise von Cunis vorgeschlagen, indem Objektdeskriptoren, Mengendeskriptoren und Teilmengendeskriptoren definiert werden (Cunis 1991). Auch in der Erweiterung der Feature-Modellierung der Softwaretechnik werden solche Kardinalitäten spezifiziert (Czarnecki, Helsen et al. 2004). Die Kardinalität definiert somit, aus wie vielen der verbundenen Komponenten sich eine Dienstleistungskomponente zusammensetzt. Demzufolge wird die Kardinalität an die Verbindungsknoten definiert. Die bislang definierte Semantik der unterschiedlichen Verbindungsknoten wird durch die Spezifikation der Kardinalitäten redundant. Für eine bessere Übersichtlichkeit und für die Beibehaltung unterschiedlicher Ausdrucksstärken werden jedoch beide Konzepte (Knotentypen und Kardinalitäten) beibehalten.

Für jeden Verbindungsknoten können die Kardinalitäten in folgenden Formen definiert werden:

1. Wertepaare: Mit einer Menge von Wertepaaren werden Wertebereiche von Kardinalitäten definiert (bspw. „zwischen 3 und 6“ oder „zwischen 8 und 12“). Bei der Definition eines exakten Wertes würde ein Wertepaar mit zwei identischen Werten spezifiziert werden.
2. Defaultwert: Sollte sich aus Erfahrungswerten oder aufgrund sonstiger Restriktionen ein Standardwert für einen bestimmten Kardinalitätswert etabliert haben, dann kann dieser durch einen Defaultwert angegeben werden.

Die Abbildung 8 zeigt beispielhaft, wie Kardinalitäten und Defaultwerte für Verbindungsknoten definiert werden können.

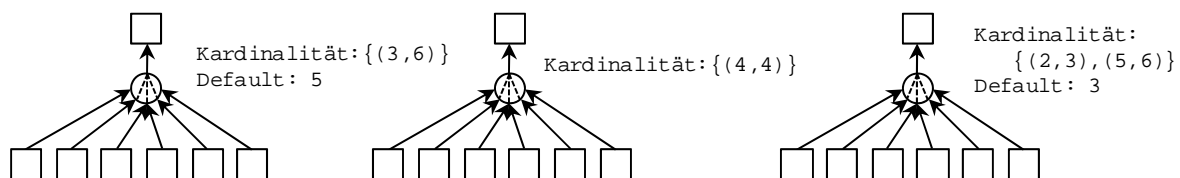


Abbildung 8: Kardinalitätsbeispiele

Aufgrund der Redundanz zwischen den Typen von Verbindungsknoten und den Kardinalitäten ist darauf zu achten, dass diese nicht widersprüchlich sind. So darf beispielsweise ein Exklusives-Oder-Knoten nur die Kardinalität (1,1) und den Defaultwert (1) besitzen. In diesem Falle ist eine Angabe der Kardinalität obsolet.

## 5.2.2 Feature-Konfiguration

Der spezifizierte Konfigurationsgraph ermöglicht die Darlegung der Zusammensetzung einer Dienstleistungskomponente aus anderen Komponenten. Des Weiteren besitzen einzelne Komponenten Eigenschaften (Abschnitt 4). Auch diese Eigenschaften können im Rahmen einer Konfiguration angepasst werden, indem diese in unterschiedlichen Ausprägungen für eine Komponente definiert werden (bspw. unterschiedliche Orte oder unterschiedliche Ressourcen).

Zur Repräsentation dieser Eigenschaftskonfiguration werden neue Knoten für den Graphen festgelegt. Zunächst wird der Eigenschaftsknoten definiert, der an eine Komponente angehängt wird. Hierbei werden nur Eigenschaften spezifiziert, die im Rahmen der Konfiguration eine Auswahlmöglichkeit aus unterschiedlichen Werten erlauben.

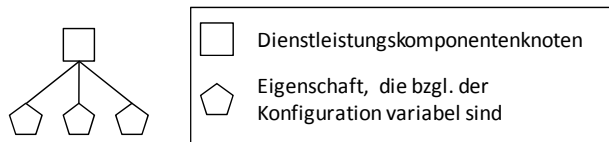


Abbildung 9: Eigenschaftsknoten im Konfigurationsgraphen

Um Eigenschaften (bspw. „Farbe“) im Rahmen der Konfiguration konkretisieren zu können (bspw. „grün“), müssen deren auswählbare Wertemengen (bspw. „grün“, „gelb“, „blau“) spezifiziert werden (Abbildung 10).

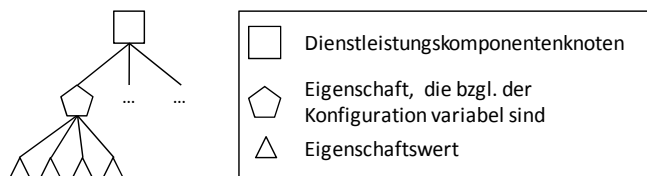


Abbildung 10: Knoten für Eigenschaftswerte

### 5.2.3 Abhängigkeitsregeln

Die im bisherigen Graphen definierbaren Abhängigkeiten zwischen Komponenten oder deren Eigenschaften hängen unmittelbar mit der Struktur des Graphen zusammen. Abhängigkeiten, die sich nicht im Graphen abbilden lassen (bspw. die Abhängigkeiten zwischen zwei Komponenten, die in unterschiedlichen Ästen aufgehängt sind), müssen jedoch ebenfalls abgebildet werden können (Abbildung 11), um Aussagen treffen zu können wie bspw. „Wenn Komponente A gewählt wurde, dann muss auch Komponente F gewählt werden.“. Das gleiche gilt auch für die Abhängigkeiten zwischen einzelnen Eigenschaftsausprägungen.

Für eine umfassende und komplexe Abhängigkeitsdarlegung bedarf es ausdrucksmächtiger Methoden wie bspw. der Aussagenlogik (Puppe 1991). Da diese in ihrer Anwendung jedoch relativ komplex ist, ist für eine überschaubare Anzahl von Abhängigkeiten auch die Spezifikation von vordefinierten Regeln ausreichend. Diese vordefinierten Regeln sind hierbei konform zu Methoden wie bspw. der Aussagenlogik zu halten, um einerseits die Konsistenz zu wahren und andererseits die wohldefinierte Erweiterung durch Erstellung weiterer Regeln zu ermöglichen.

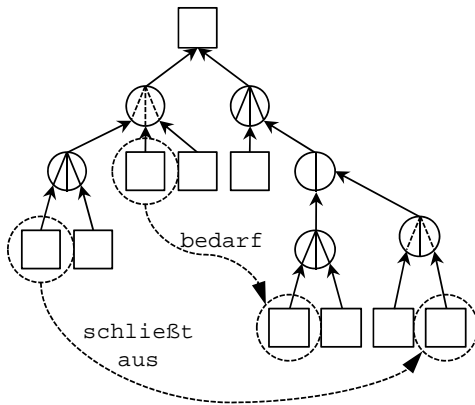


Abbildung 11: Beispielhafte Visualisierung von Abhängigkeiten

Für die Spezifikation von Abhängigkeiten lassen sich zunächst vier Regeln definieren, die ein breites Spektrum benötigter Abhängigkeitsaussagen abdecken. Die nachfolgende Tabelle legt die Regeln, den entsprechenden Ausdruck in der Aussagenlogik sowie die nähere Erläuterung der jeweiligen Regel dar.

<i>Regel</i>	<i>Aussagenlogik</i>	<i>Erläuterung</i>
<i>a_führt_zu_b</i>	$a \Rightarrow b$	Wenn eine Komponente a im Rahmen einer Konfiguration gewählt wird, dann muss auch Komponente b gewählt werden.
<i>nicht_a_führt_zu_nicht_b</i>	$\neg a \Rightarrow \neg b$	Wenn eine Komponente a nicht gewählt wurde, dann darf auch Komponente b nicht gewählt werden.
<i>a_und_b_schließen_sich_aus</i>	$(a \Rightarrow \neg b) \wedge (b \Rightarrow \neg a)$	Wenn eine Komponente a gewählt wird, dann darf Komponente b nicht gewählt werden (et vice versa).
<i>nicht_a_führt_zu_b</i>	$\neg a \Rightarrow b$	Wenn Komponente a nicht gewählt wird, dann muss Komponente b gewählt werden.

Abbildung 12: Regeln zur Darlegung von Abhängigkeiten zwischen Komponenten

Da es sich bei Dienstleistungskomponenten um Abläufe handelt, bedarf es neben der Darlegung der logischen Abhängigkeit auch der Wiedergabe der zeitlichen Abhängigkeiten, um Restriktionen bzgl. des Ablaufs gewählter Komponenten definieren zu können. Hierbei bietet sich die Verwendung eines deklarativen Ansatzes (Aalst and Pesic 2006) an, um die Flexibilität des Konfigurationsgraphen beizubehalten. Anstelle ganzer Prozesse werden nur Regeln für den letztendlichen Ablauf definiert. Auch hierbei kann auf existierende Beschreibungsmethoden wie bspw. die lineare temporale Logik (Dierks and Olderog 2000) zurückgegriffen werden.

## 6 Literatur

- Aalst, M. W. P. v. d. and M. Pesic (2006). Specifying, Discovering, and Monitoring Service Flows: Making Web Services Process-Aware. BPM Center Report, BPM Center.
- Akkermans, H., Z. Baida, et al. (2004). "Value Webs: Using Ontologies to Bundle Real-World Services." IEEE Intelligent Systems **19**(4): 57-66.
- Bäcker, M. and L. Herzog (2004). Modulares Dienstleistungskonzept eines mittelständischen Maschinenbauunternehmens. Dienstleistungsorientierte Geschäftsmodelle im Maschinen- und Anlagenbau: Vom Basisangebot bis zum Betreibermodell. H. Meier. Berlin, Springer: 125-140.
- Baida, Z., J. Gordijn, et al. (2003). Service Ontology, Vrije Universität Amsterdam.
- Baldwin, C. Y. and K. B. Clark (1997). "Managing in an Age of Modularity." Harvard Business Review **75**: 84-93.
- Balzert, H. (2000). Lehrbuch der Software-Technik - Software- Entwicklung / Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung. Heidelberg, Berlin, Spektrum Akademischer Verlag.
- Balzert, H. (2000). Lehrbuch der Software-Technik - Software Entwicklung. Heidelberg, Berlin, Spektrum Akademischer Verlag.
- Becker, S. and S. Overhage (2003). Stücklistenbasiertes Komponenten-Konfigurationsmanagement. Tagungsband des 5. Workshops Komponentenorientierte betriebliche Anwendungssysteme (WKBA 5). K. Turowski. Augsburg: 17-32.
- Bettag, U. (2001). "Web-Services." Informatik Spektrum **24**(5): 302-304.
- Blohm, H., T. Beer, et al. (1997). Produktionswirtschaft. Berlin, Verlag Neue Wirtschafts-Briefe.
- Böhm, T., M. Junginger, et al. (2002). Modular Service Architectures: A Concept and Method for Engineering IT Services. 36th Annual Hawaii International Conference on System Sciences, Hawaii (USA).
- Böttcher, M. (2009). Architektur integrierter Dienstleistungssysteme - Konzeption, Metamodell und technikraumsspezifische Konkretisierung. Institut für Informatik. Leipzig, Universität Leipzig. **Dr. rer. nat.**
- Böttcher, M. and K.-P. Fähnrich (2009). Service Systems Modelling. Proceedings First International Symposium on Services Science (ISSS'09). R. Alt, K.-P. Fähnrich and B. Franczyk. Berlin, Logos.
- Brown, A., S. Johnston, et al. (2002). Using Service-Oriented Architecture and Component-Based Development to build Web Service Applications. A Rational Software White Paper, Rational Software Corporation.
- Brusoni, S. and A. Prencipe (2001). "Unpacking the Black Box of Modularity: Technologies, Products and Organizations." Industrial and Corporate Change **10**(1): 197-315.
- Burianek, F., C. Ihl, et al. (2007). Typologisierung hybrider Produkte - Ein Ansatz basierend auf der Komplexität der Leistungserbringung. Arbeitsberichte des Lehrstuhls für Betriebswirtschaftslehre - Information, Organisation und Management. R. Reichenwald. München, Technische Universität München.
- Burr, W. (2002). Service Engineering bei technischen Dienstleistungen. Wiesbaden, Deutscher Universitäts-Verlag.
- Corsten, H., K.-M. Dresch, et al. (2006). Modular service production - A coordination-focused analysis. IFSAM VIIIth World Congress 2006, Berlin, Germany.
- Corsten, H., K.-M. Dresch, et al. (2007). Gestaltung modularer Dienstleistungsproduktion. Wertschöpfungsprozesse bei Dienstleistungen. M. Bruhn and B. Stauss. Wiesbaden, Gabler.
- Cunis, R. (1991). Modellierung technischer Systeme in der Begriffshierarchie. Das PLAKON-Buch - Ein Expertensystemkern für Planungs- und Konfigurierungsaufgaben in technischen Domänen. R. Cunis, A. Günter and H. Strecker. Berlin et al., Springer: 58-76.
- Cunis, R. and A. Günter (1991). PLAKON - Übersicht über das System. Das PLAKON-Buch - Ein Expertensystemkern für Planungs- und Konfigurierungsaufgaben in technischen Domänen. R. Cunis, A. Günter and H. Strecker. Berlin et al., Springer: 37-57.
- Czarnecki, K. and U. Eisenecker (2000). Generative Programming: Methods, Tools and Application, Addison-Wesley Professional.
- Czarnecki, K., S. Helsen, et al. (2004). Staged Configuration Using Feature Models. Software Product Lines - Third International Conference. Boston, MA, USA, Springer.
- Dabic, M. (2005). "Produkte nach Maß für jedermann?" Transfer Werbeforschung und Praxis(1&2): 26-30.
- Dierks, H. and E.-R. Olderog (2000). "Temporale Spezifikationslogiken." at - Automatisierungstechnik **48**(0): 7-10.
- Frank, U. (1999). "Component Ware - Software-technische Konzepte und Perspektiven für die Gestaltung betrieblicher Informationssysteme." Information Management & Consulting **14**(2): 11-18.
- Gabler (2000). Gabler-Wirtschafts-Lexikon. Wiesbaden, Gabler.

- Gao, T., H. Ma, et al. (2006). "A Repository for Component-Based Embedded Software Development." International Journal of Software Engineering & Knowledge Engineering **16**(4): 523-552.
- Genba, K., H. Ogawa, et al. (2005). "Quantitative Analysis of Modularization in the Automobile and PC Industries." Technology Analysis & Strategic Management **17**(2): 231-245.
- Göpfert, J. and M. Steinbrecher (2000). "Modulare Produktentwicklung leistet mehr." Harvard Business Manager **3**: 20-31.
- Gräßler, I. (1999). Kundenindividuelle Massenproduktion. Berlin, Heidelberg, New York, Springer.
- Grieble, O. (2004). Modellgestütztes Dienstleistungsbenchmarking. Rechts- und Wirtschaftswissenschaftliche Fakultät. Saarbrücken, Universität des Saarlandes. **Doktor der Wirtschaftswissenschaften: XXVI**, 287 S.
- Hermesen, M. (2000). Ein Modell zur kundenindividuellen Konfiguration produktnaher Dienstleistungen - Ein Ansatz auf Basis modularer Dienstleistungsobjekte. Fakultät für Maschinenbau. Bochum, Ruhr-Universität Bochum.
- Herrmann, K. and R. Klein (2004). Methodenbasierte Visualisierung von Dienstleistungen. Computer Aided Service Engineering - Informationssysteme der Dienstleistungsentwicklung. A.-W. Scheer and D. Spath. Berlin et al., Springer. **1**: 93-119.
- Herrmann, K., R. Klein, et al. (2006). Computer Aided Service Engineering - Konzeption eines Service Engineering Tools. Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. H.-J. Bullinger and A.-W. Scheer. Berlin et al., Springer: 649-678.
- ISO (2004). ISO 8601 - Data elements and interchangeformats - Information interchange - Representation of dates and times. International Standards. I. O. f. Standardization, International Organization for Standardization.
- Leukel, J. and V. Schmitz (2002). Modellierung von Preisinformationen in elektronischen Produktkatalogen. GI-Arbeitstagung Modellierung 2002, Tutzing, Gesellschaft für Informatik.
- Mehlau, J. I. and A. Wimmer (2002). Produktmodelle im Finanzdienstleistungssektor - Entwicklung eines objektorientierten Meta-Modells. Regensburger Diskussionsbeiträge zur Wirtschaftswissenschaft. Regensburg, Wirtschaftswissenschaftliche Fakultät.
- Meier, H., W. E. Maßberg, et al. (2003). Kundenindividuelle Services auf Basis modularer Dienstleistungen. Marchance Individualisierung. G. Reinhart and M. F. Zäh. Berlin et al., Springer.
- O'Sullivan, J. (2006). Towards a Precise Understanding of Service Properties. Faculty of Information Technology, Queensland University of Technology. **Doctor of Philosophy**.
- o.V. (2001). Duden Informatik. Mannheim et al., Duden Verlag.
- Peters, L. and H. Sadin (2000). "IT and the mass customization of services: the challenges of implementation." International Journal of Information Management **20**: 103-119.
- Pfeiffer, D. and A. Winkelmann (2007). "Ansätze zur Wiederversendung von Software im Rahmen der Softwareindustrialisierung am Beispiel von Softwarekomponenten, service-orientierten Architekturen und modellgetriebenen Architekturen." Wirtschaftsinformatik **49**(3): 208-216.
- Piller, F. T. and C. M. Stotko (2003). Vom Faustkeil über das Fließband zur Mass Customization? Mass Customization und Kundenintegration - Neue Wege zum innovativen Produkt. F. T. Piller and C. M. Stotko. Düsseldorf, Symposion.
- Piller, F. T. and C. M. Stotko (2003). Wie ein Mass-Customization-System entwickelt wird. Mass Customization und Kundenintegration - Neue Wege zum innovativen Produkt. F. T. Piller and C. M. Stotko. Düsseldorf, Symposion.
- Pine II, J. B. (1999). Mass Customization: The Frontier in Business Competition. Boston, MA, Harvard Business School Press.
- Puppe, F. (1991). Einführung in Expertensysteme. Berlin et al., Springer Verlag.
- Russel, S. and P. Norvig (2004). Künstliche Intelligenz - Ein moderner Ansatz. München, Pearson Education.
- Sametinger, J. (1997). Software Engineering with Reusable Components. Berlin, Heidelberg, et al., Springer.
- Schramm, J., J. and K. Pallentien (2004). Entwicklung und Handhabung modularer Dienstleistungsbaukästen. Dienstleistungsorientierte Geschäftsmodelle im Maschinen und Anlagenbau - Vom Basisangebot bis zum Betreibermodell. H. Meier. Berlin, Heidelberg, Springer.
- Starr, M. K. (1965). "Modular Production - A New Concept." Harvard Business Review **43**(6): 131-142.
- Stormer, H. (2007). "Kundenbasierte Produktkonfiguration." Informatik Spektrum **30**(5): 322-326.
- Szyperski, C. (2002). Component software - beyond object-oriented programming. London et al., Addison-Wesley.
- Teschl, G. and S. Teschl (2006). Mathematik für Informatiker - Band 2: Analysis und Statistik. Berlin, et al., Springer.

- Thomas, O. and A.-W. Scheer (2006). Customizing von Dienstleistungsinformationssystemen. Service Engineering - Entwicklung und Gestaltung innovativer Dienstleistungen. H.-J. Bullinger and A.-W. Scheer. Berlin et al., Springer: 677-718.
- Thomas, O., P. Walter, et al. (2008). "Product-Service Systems: Konstruktion und Anwendung einer Entwicklungsmethodik." Wirtschaftsinformatik **50**(3): 208-219.
- Vazsonyi, A. (1962). Die Planungsrechnung in Wirtschaft und Industrie. Wien et al., Oldenbourg.
- Zahn, E. and U. Schmidt (1996). Produktionswirtschaft I: Grundlagen und operatives Produktionsmanagement. Stuttgart, Lucius and Lucius.