

# Mass Customization für IT-Dienstleistungen realisieren

Dipl.-Ing. univ. Tonio Grawe  
Prof. Dr.-Ing. habil. Dipl.-Math. Klaus-Peter Fähnrich

Universität Leipzig  
Institut für Informatik, Abt. für Anwendungsspezifische Informationssysteme

**Abstract:** Die Entwicklung und kontinuierliche Innovation von IT-Dienstleistungen, beispielsweise der Betrieb von Anwendungssystemen und entsprechender Anwender-Support, ist heute noch wenig systematisiert. Oft sind die IT-Dienstleistungen monolithisch und relativ starr. Wir stellen einen methodischen Ansatz vor, wie komplexe IT-Dienstleistungen aus Komponenten aufgebaut werden können. Eine solche Plattformstrategie verspricht, im Sinne von Mass Customization kundenspezifische Dienstleistungen anbieten zu können. Die systemgestützte Konfiguration neu komponierter Dienstleistungen enthält nicht-triviale Probleme, die durch Methoden der Wissensrepräsentation gelöst werden können.

## 1 IT-Dienstleistungen

Innovation entsteht bei IT-Betreibern, indem das Dienstleistungsangebot kontinuierlich an den Anforderungen der Kunden ausgerichtet wird. Das strategische Portfoliomanagement wacht darüber, dass die angebotenen Dienstleistungen aufeinander abgestimmt sind, dagegen ist das Produktmanagement für den Lebenszyklus der einzelnen Dienstleistungen verantwortlich. Soweit unterscheidet sich das Geschäft eines IT-Dienstleisters nicht von anderen Branchen.

Die Besonderheit bei der Entwicklung von Dienstleistungen liegt in der mangelnden Greifbarkeit. "Den Betrachtungsgegenstand begriffen bzw. wahrgenommen zu haben, ist jedoch eine Grundvoraussetzung für die effiziente Gestaltung von Dienstleistungen" [ML02]. Im Fall von IT-Services wird aus diesem Grund fälschlicherweise oftmals das Greifbare, nämlich die physikalische Infrastruktur, mit der Dienstleistung gleichgesetzt.

Dienstleistungen lassen sich durch vier Komponenten beschreiben: (1) das Produktmodell, das ist die Spezifikation der Dienstleistung, (2) ein Prozessmodell, (3) ein Ressourcenmodell, und (4) ein Marketingkonzept. Letztgenanntes beinhaltet insbesondere Kundeninteraktionsmodelle und wird damit der Forderung aus [Br02] gerecht, dass speziell bei Dienstleistungen der Kunde stärker in den Erbringungsprozess eingebunden werden muss. Informationstechnologische Infrastruktur zählt neben "Human Resources" und Betriebsmitteln zu den Ressourcen. Sie ist offensichtlich zwar ein wichtiger, aber doch nur ein Teil der Dienstleistung.

Der von uns vorgeschlagene Ansatz, die IT-Dienstleistung greifbar zu machen, ist die systemgestützte Modellierung aller relevanten Bestandteile der IT-Dienstleistung.

## **2 Komponentenbasierte IT-Dienstleistungen**

### **2.1 Plattformstrategie**

Der Begriff "Plattformstrategie" ist aus der Automobilindustrie bekannt. Gemeint ist damit eine Standardisierung von Komponenten. Dieser Ansatz, der die Automobilbranche hinsichtlich Effizienz und Kundenorientierung revolutioniert hat, kann in den IT-Betrieb übernommen werden. Die Produkte (Autos) sind IT-Dienstleistungen, die dem Kunden verkauft werden. Die Komponenten (Motor, Fahrwerk) sind Teilleistungen. Das Ziel ist nun, durch Standardisierung eine Teilleistung möglichst nur einmal zu erbringen und diese in möglichst viele Produkte einfließen zu lassen.

### **2.2 Vorgehensweise zur Einführung komponentenbasierter IT-Services**

Um eine Plattformstrategie erfolgreich einzuführen, schlagen wir ein Vorgehen in fünf Schritten vor:

1. **Dekomposition:** Komplexe IT-basierte Dienstleistungen werden durch Reverse-Engineering in ihre Komponenten zerlegt. Durch die mehrstufige Dekomposition wird eine komplexe Dienstleistung in Teilleistungen zerlegt und schließlich werden ihr Ressourcen, sei es Personal oder Infrastruktur, zugeordnet. Je höher die Granularität der Komponenten ist, desto genauer können Daten aggregiert werden und desto mehr Ansatzpunkte finden sich, die Komponenten wiederzuverwenden.
2. **Standardisierung:** Komponenten mit gleicher Funktionalität werden vereinheitlicht. Dazu wird eine Klassifizierung der Komponenten durch Abstraktion durchgeführt. Eine Klasse kann dabei als Spezifikation für die Dienstleistung verstanden werden. Die Klassifizierung hilft somit, die Gemeinsamkeiten oder auch Unterschiede der Dienstleistungskomponenten zu dokumentieren.
3. **Taxonomie:** Anwendung einer Taxonomie für IT-Dienstleistungen bzw. Ableitung einer allgemeingültigen Taxonomie aus der erstellten Klassifizierung. Damit wird den Komponenten implizit eine Semantik verliehen. Andere Marktteilnehmer in der Wertschöpfungskette können so die Funktion der Dienstleistungskomponente einfacher verstehen.
4. **Wissensrepräsentation:** Abbilden des Konfigurationswissens durch Modellierung der Dienstleistungskomponenten. Anhand der Semantik können Abhängigkeiten generell, also unabhängig von der Instanz, modelliert werden. Dies ist die Grundlage für das computergestützte Entwerfen neuer komplexer Dienstleistungen mit Hilfe von Produktkonfiguratoren.
5. **Partnerintegration:** Die Dienstleistungskomponenten müssen in Wertschöpfungsnetzen gehandelt werden können. Es bietet sich hier die Automatisierung durch e-Business an. Wir schlagen vor, die Technologiestandards von Web-Services zu verwenden: UDDI für Leistungskataloge, WSDL zur Beschreibung der Dienstleistungsprodukte und BPEL für die Beschreibung der Prozesse zwischen Anbieter und Kunde. Einen Überblick über diese Technologien gibt [Gr03].

## 2.3 Komposition komplexer Dienstleistungen

Die oben beschriebene Vorgehensweise enthält eine Bestandsaufnahme und liefert als Ergebnis einen Katalog von Dienstleistungskomponenten, die sorgfältig dokumentiert sind. Die Komponenten dienen nun als Basis für die Entwicklung neuer Dienstleistungen, welche auf den bestehenden Komponenten aufbauen. Diesen Vorgang bezeichnen wir als die Komposition von Dienstleistungen.

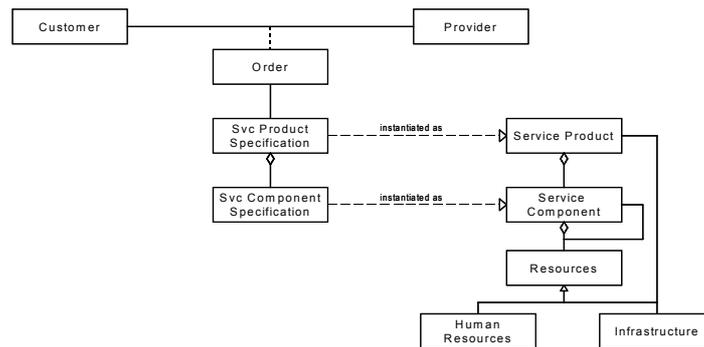


Abb. 1: Ontologie der Domäne

Die Abbildung stellt die von uns verwendete Ontologie der Domäne dar, die in Anlehnung an [RSF02] entstanden ist. Die Telekommunikationsbranche, die wie IT-Betreiber auch Dienstleister ist, aber gemeinhin schon professioneller aufgestellt ist, hat komponentenbasierte Dienstleistung zu einer charakteristischen Eigenschaft ihrer selbst gemacht.

## 3 Mass Customization

Beim Mass Customization wird der Kunde in den Entwicklungs- bzw. Design-Prozess eingebunden, wodurch sich die Anforderungen des Kunden viel besser in dem Produkt wiederfinden [Zi01]. Während die Idee dazu bereits auf [To70] zurückgeht, ist deren Realisierung erst durch e-Business effektiv möglich geworden.

Es besteht jedoch die Gefahr, dass eine Vielzahl von Konfigurationsmöglichkeiten den Kunden überfordert. Es entsteht "mass confusion" [Te03]. Ein IT-Dienstleister hat bis zu 1000 Komponenten im Betrieb. Nur ein kleiner Teil der möglichen Permutationen ist sinnvoll und erlaubt. Der Freiheitsgrad bei der Konfiguration muss also eingeschränkt werden. Die Grenze zwischen Mass Customization und Mass Confusion ist heute noch nicht bekannt und wird in [FP02] als Forschungsgebiet festgelegt.

Das von uns vorgeschlagene System führt den Anwender durch die Konfiguration der Dienstleistung. Wir folgen einem Vorschlag aus [Ha98] und unterscheiden zwischen "high level configuration", wobei die Konfiguration interaktiv vom Anwender vorge-

nommen wird, und "low level configuration", was automatisch vom System konfiguriert wird.

### **3.1 Semantisches Wissen**

Eine erste Einschränkung im Freiheitsgrad der Konfiguration ist durch eine Produktspezifikation vorgegeben, in der das Produktmanagement festlegt, welche Funktionen das Produkt erfüllen soll. Die Funktion einer jeden Dienstleistungskomponente wird durch ihre Semantik beschrieben. Alle Komponenten sind durch eine Taxonomie klassifiziert. Durch die Einordnung in diese Taxonomie ist das semantische Wissen erschlossen. Die Taxonomie ist einerseits allgemein genug, um die Funktion einer Komponente verständlich zu beschreiben, andererseits bringt sie die Flexibilität mit, erweiterbar zu sein für jegliche denkbare neue Dienstleistungskomponente.

Durch die Produktspezifikation ist festgelegt, aus welchen Klassen IT-Dienstleistungskomponenten für eine kundenspezifische Konfiguration ausgewählt werden können.

### **3.2 Constraints-Technologie**

Eine weitere Einschränkung ergibt sich für die Konfiguration dadurch, dass es Komponenten gibt, die sich gegenseitig ausschließen, weil charakteristische Eigenschaften nicht zusammenpassen. Beispielsweise kann eine Komponente nicht einem Produkt zugeordnet werden, wenn die Betriebszeit der Komponente geringer als die des Produkts ist. Solche Bedingungen können im System durch Constraints, das sind Gleichungen oder Ungleichungen der Prädikatenlogik, hinterlegt werden.

Im Automobilbereich, in dem der Konfigurationsumfang noch größer als im IT-Bereich ist, wurde die Erfahrung gemacht, dass die Fehlerrate bei den Bedingungen deutlich gesenkt werden kann, wenn die Regeln in natürlicher Sprache verständlich sind [SKK03]. Es muss also ein Kompromiss aus formalisierter und natürlicher Sprache gefunden werden. Goltz führt in [Go99] aus, dass nicht die Abbildung der Bedingungen, sondern das Finden von Problemlösungen unter Berücksichtigung der Constraints innerhalb angemessener Zeit die Herausforderung ist. Außer den Sprachen, die speziell für die Constraints-Programmierung entwickelt wurden, gibt es heute für die gängigen Programmiersprachen entsprechende Problemlöser<sup>1</sup>.

Constraints werden auch eingesetzt, um Abhängigkeiten zwischen Komponenten abzubilden. Bedingt eine Komponente eine andere, so kann das System automatisch ohne Anwenderinteraktion die zusätzliche Komponente konfigurieren.

## **4 Anwendungsbezogene Diskussion**

Wir haben eine Vorgehensweise vorgestellt, wie IT-Betreiber komponentenbasierte IT-Dienstleistungen einführen können. Komplexe Dienstleistungen wie beispielsweise die

---

<sup>1</sup> Etwa die Java Constraint Library als Open-Source-Software oder Jsolver des französischen Anbieters ILOG.

Bereitstellung eines ausgestatteten Arbeitsplatzes für einen Mitarbeiter werden in wiederverwendbare Komponenten zerlegt. In diesem Beispiel könnte dies der Betrieb eines Desktop-PCs, Betrieb von Netzwerken, Betrieb von Anwendungen im Rechenzentrum und telefonischer Support sein.

Der komponentenbasierte Ansatz verspricht Innovation, weil sich durch die Wiederverwendung neue Produkte schnell entwickeln lassen, Effizienz, weil Ressourcen besser ausgelastet werden können, und Kundenorientierung, weil durch Mass Customization auf die individuellen Anforderungen des Kunden eingegangen werden kann. Für den IT-Dienstleister sind dies wertvolle Wettbewerbsvorteile.

Die laufenden Forschungsarbeiten konzentrieren sich auf die Konfiguration von IT-Dienstleistungsprodukten. Dabei muss der Anwender systemgestützt aus einer gegebenen Menge von Komponenten sinnvolle und umsetzbare Produkte komponieren können. Die Arbeiten profitieren von den Erfahrungen, die im Maschinenbau und im Automobilbereich mit Produktkonfiguration gemacht wurden und adaptieren diese auf die speziellen Anforderungen der IT-Dienstleistungen. Kontinuierlich erbrachte Dienstleistungen unterscheiden sich grundsätzlich von diskreten Gütern. Es müssen Attribute, die unabhängig von der jeweiligen Dienstleistungskomponente sind (etwa Verfügbarkeitsgrad, Kosten, Wartungsintervalle, Wiederinstandsetzungszeit), und welche, die spezifisch für eine bestimmte Dienstleistung oder eine Kategorie von Dienstleistungen sind, berücksichtigt werden. Beispiele für letztes sind die Warteschleifendauer bei einem Call Center, Antwortzeiten bei verteilten Applikationen oder die "Packet Loss Ratio" bei einem IP-Netz. Der Definition einer allgemeingültigen und erweiterbaren Taxonomie für IT-Dienstleistungskomponenten kommt aus diesem Grund besondere Bedeutung zu.

## Literaturverzeichnis

- [Br02] Bruhn, M.: Integrierte Kundenbindung: Implementierung einer kundenorientierten Unternehmung. Wiesbaden, 2002.
- [FP02] Franke, N.; Piller, F. T.: Configuration Toolkits for Mass Customization. Arbeitsbericht Nr. 33 des Lehrstuhls für Allgemeine und Industrielle Betriebswirtschaftslehre, TU München, 2002.
- [Go99] Goltz, H.-J.: Problemlösen mit constraintbasierter Programmierung. Arbeitspapier der GMD, 1999.
- [Gr03] Grawe, T.: Geschäftsstrategie für IT-Dienstleister (Teil 2): Effizienz. In: IT Management, Heft 8, 2003.
- [Ha98] Haag, A.: Sales configuration in business processes. IEEE Intelligent Systems, Vol. 13, No. 4, July/August 1998.
- [ML02] Meiren, T.; Liestmann, V. (Hg): Service Engineering in der Praxis. Fraunhofer IRB Verlag, Stuttgart, 2002.
- [RSF02] Reilly, J.; Strasser, J.; Faurer, C. et. al.: Shared Information/Data (SID) Model, Version 2.0. TeleManagement Forum, 2002.
- [SKK03] Sinz, C.; Kaiser, A.; Küchlin, W.: Formal Methods for the Validation of Automotive Product Configuration Data. University of Tübingen, 2003.
- [Te03] Teresko, J: Mass Customization or Mass Confusion. Industry Week, Vol. 243, No. 12, S. 45-48.
- [To70] Toffler, A.: Future Shock. Bantem Books, New Nork, 1970.
- [Zi01] Zipkin, P. H.: The Limits of Mass Customization. Sloan Management Review, 2001, Vol. 42, S. 81-87.