

# Fertigung kundenindividueller Geschäftsprozesse mittels einer Process Factory

Andrej Werner

Institut für Wirtschaftsinformatik  
Universität Leipzig  
Marschnerstr. 31  
04109 Leipzig  
a.werner@wifa.uni-leipzig.de

**Abstract:** Der Ruf der Kunden nach individuellen Anwendungssystemlösungen zu Preisen von Standardsoftwarelösungen und zu deutlich niedrigeren Einführungs- und Anpassungskosten ist deutlich hörbar. Die Unternehmen richten ihre individuellen Geschäftsprozesse nach den variierenden Kundenanforderungen aus und die daraus resultierenden Kern- und Unterstützungsprozesse variieren in den Abläufen oder Ressourcen. Im Kampf um bessere Produkte und Dienstleistungen grenzen sich so die Unternehmen voneinander ab. Anwendungssysteme unterstützen diese unternehmensindividuellen Geschäftsprozesse und bewältigen die Komplexität der durchzuführenden Arbeitsschritte. Wer am effektivsten seine Geschäftsprozesse organisieren und diese effizient mit Anwendungssystemen umsetzen kann, erreicht signifikante Wettbewerbsvorteile am Markt. Daher besteht die wirtschaftliche Notwendigkeit Techniken zu entwickeln, die die kundenindividuelle Fertigung von Anwendungssystemen zu niedrigeren Produkt- und Anpassungskosten ermöglichen. Das Konzept der Process Factory beschreibt einen Teil der notwendigen Techniken.

## 1 Einleitung

In den Unternehmen fand der Wechsel von funktionsorientierter Betrachtung hinzu prozessorientierter Betrachtung von Organisationen statt. Bei der Untersuchung von Architekturen betrieblicher Anwendungssysteme ist ein Wechsel zu prozessorientierten Softwaresystemen (z.B. [SAPN]) erkennbar. Der Übergang wurde durch die Thematisierung von Workflow-Techniken und –Systemen eingeleitet und durch die Entwicklung von Techniken zur Beschreibung und Ausführung von Geschäftsprozessen zur Systemlaufzeit (z.B. [BPEL]) verstärkt. Mass Customization steht für eine Synthese aus Massenfertigung und der Fertigung kundenindividueller Endprodukte mit der Konzentration auf den Kunden, der in seinen Augen ein individuelles Produkt erhält [Pi94]. Daraus folgt als Ziel, die Entwicklung von Anwendungssystemen mit so viel Variabilität zu planen, dass fast jeder Kunde genau die von ihm gewünschte Lösung finden kann. Das Thema Variabilität von Geschäftsprozessen (GP) wird bereits in der Literatur diskutiert und ist daher nicht neu (vgl. [Sc98, La97, Re97, Kö97, Sp04 et al.]). Das Konzept einer Process Factory basiert auf der Zusammenführung verschiedenster Techniken aus dem Domain-, Business- und Software-Engineering.

Es stellt einen Ansatz zum effizienten Management der Variabilität von Geschäftsprozessen in Organisationen und Anwendungssystemen dar und beschreibt ein softwaregestütztes Werkzeug, das die ingenieurmäßige Konstruktion, Bereitstellung von Geschäftsprozessfamilien-Modellen und die Fertigung von prozessorientierten Software- oder Organisation-Assets nach Kundenanforderungen für die Verwendung in Softwarelösungen oder für die Dokumentation der Unternehmensarchitektur unterstützt [We05]. Eine Kombination von Process- und Software Factories [GS04] ermöglicht die kundenindividuelle Entwicklung sowie Anpassung von Anwendungssystemen mit der Passgenauigkeit von Individualsoftware und den Kostenvorteilen einer Standardsoftware.

## **2 Process Factory**

### **2.1 Prinzip**

Der Begriff der Geschäftsprozessfamilie wird vom Begriff der Systemfamilie abgeleitet. Eine Geschäftsprozessfamilie (GPF) einer Domäne repräsentiert eine Gruppe von GP, die auf einer gemeinsamen Menge dynamischer und statischer Prozessbausteine basieren. Ein Prozessbaustein (GPB) stellt die kleinste unzerlegbare logische Einheit eines GP dar und wird von einer Ressource realisiert. Die einheitlich verwendeten GPB bilden die Gemeinsamkeiten der GPF. Die Variabilität der Familie kann in dynamischen oder statischen GPB definiert sein. Durch unterschiedliche dynamische GPB (Ablauffolge) kann der Funktionsumfang der Prozessausprägungen sowie der Ablauf des GP variierend gestaltet werden.

Bei der GPF-Entwicklung steht die Konstruktion eines Merkmalmodells für die strukturierte Sicht einer GPF sowie einer verhaltensorientierten Sicht eines generischen Ablaufes einer GPF zur Aufgabe. Das bedeutet, dass für die Beschreibung einer GPF mindestens zwei Modell-Sichten notwendig sind. Alle potentiellen GPB einer Domäne sollen identifiziert und dokumentiert werden. Dazu müssen Domäneninformationen und -anforderungen, die die Kombinationsvielfalt der GPB in Konfigurationsregeln begrenzen, einer Referenz-Organisation oder eines Anwendungssystems mit Hilfe einer Domänenanalyse untersucht werden. Die GPB beherbergen Variationspunkte, an denen die Variabilität eines GP zur GP-Assetentwicklung konkretisiert werden muss. Diese sollen bei der GPF-Entwicklung in ihren potenziellen Ausprägungen untersucht und modelliert werden. Das Ergebnis sind Informationsmodelle, die eine Menge von Geschäftsprozessvarianten beinhalten. Die Verwendung von Templates für die Beschreibung von generischen GP ermöglicht die Ersetzung (Konkretisierung) von Platzhaltern variantenreicher GPB durch kundenspezifische GPB. Hierzu müssen die Kundenanforderungen an die zu realisierenden GP mit Hilfe einer Geschäftsprozessanalyse ermittelt werden. Das Ergebnis ist eine konkrete kundenindividuelle Instanz einer GPF (als Referenz-Modell). Diese Modellinstanz stellt ein von der Anwendungssystemimplementierung unabhängiges prozessorientiertes (Software-)Asset dar. Unter einem prozessorientierten Software-Asset versteht man eine Menge von Artefakten, die die Struktur und das Verhalten von GP mit verschiedenen Sichten (je Artefakt) für einen konkreten Fall modellhaft beschreiben.

Die Prägung dieser Assets in Bezug auf eine Zielsprache, die Anwendungssysteme zur Ausführung von Prozessmodellen verwenden, soll in einem Transformationsprozess realisiert werden (Abb. 6). Hierzu sind Generatoren für bestimmte Zielsprachen notwendig. Sie übernehmen das Mapping der Elemente des kundenspezifischen GP auf die Elemente der Zielsprache. Zum Schluss erhält man Zielsprachen- (bzw. Anwendungssystem-) abhängige prozessorientierte Software-Assets. Wurden Organisation-Assets in der GP-Assetentwicklung betrachtet und anhand unternehmensbezogener Anforderungen in konkrete Unternehmensinformationsmodelle überführt, so sind diese Modelle unabhängig von herstellereigenen Modellierungssprachen. Der abschließende Schritt stellt die Anwendung von Generatoren für Zielsprachen der Modellierungswerkzeuge für die Erstellung konkreter Organisation-Assets dar.

## 2.2 Beispiel

Das folgende beispielhafte Szenario beschreibt die Anwendung einer Process Factory. Ein e-Business- Unternehmen verkauft Produkte über das Internet und führt dazu verschiedene Geschäftsprozesse aus. Im Beispiel wird der Kern-Geschäftsprozess „Verkauf“ der Domäne „Vertrieb“ und im speziellen der Teilprozess „Angebot“ detaillierter betrachtet. Das Unternehmen möchte eine prozessorientierte e-Business-Lösung für die informationstechnische Unterstützung des Verkaufsprozesses einsetzen. Sie basiert auf einem Anwendungsserver, der BPEL zur Prozessausführung einsetzt. Die verschiedenen Systemkomponenten sind als Webservice gekapselt und können von einer BPEL-Instanz angesprochen werden. Eine Beschreibung der GP, die das Anwendungssystem ausführen kann, wurde vom Systemhersteller in Form der GPF erstellt. Der generische GP „Verkauf“ variiert im Teilprozess „Angebot“. Abb. 1 und Abb. 2 stellen zwei sinnvolle Variantenausprägungen des Teilprozess „Angebot“ dar.



Abb. 1: Teilprozess Angebot - Variante 1

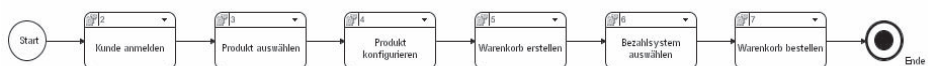


Abb. 2: Teilprozess Angebot - Variante 2

Es existiert eine endliche Anzahl potentieller kundenindividueller Ausprägungen des GP „Verkauf“. Abb.3 zeigt ein Merkmalmodell der GPF „Verkauf“ und Abb. 4 zeigt den Raum potentieller Konfigurationen für diese GPF auf. Das Anwendungssystem stellt im betrachteten Teilprozess sieben GPB der GPF „Verkauf“ bereit. Ein GPB ist optional (siehe Abb.3, GPB „Warenkorb löschen“) und zwei GPB stellen eine alternative Auswahl dar (siehe Abb.3, GPB „Adressdaten erstellen“ und GPB „Kunde anmelden“). Bei der GP-Assetentwicklung soll durch Kundenanforderungen aus der GPF „Verkauf“ ein konkreter GP „Verkauf“ konfiguriert werden. In einem weiteren Schritt soll dieser GP in ein prozessorientiertes Software-Asset transformiert werden.

Die Unternehmensanforderung an den GP „Verkauf“ lautet, dass für die Berechnung eines Produktpreises der Kunde im System angemeldet sein muss. Ausgehend von dieser Anforderung wird aus dem Konfigurationsraum die GP-Variante 2 erstellt, wie in Abb. 2 und Abb. 5 zu sehen sind.

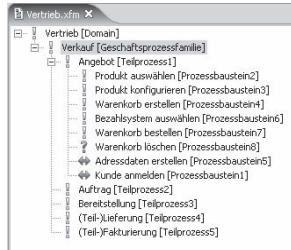


Abb.3: Merkmalmodell der GPF "Verkauf"

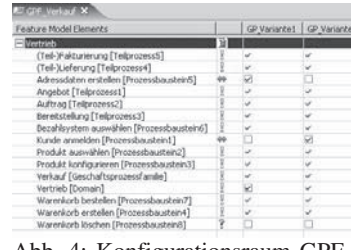


Abb. 4: Konfigurationsraum GPF "Verkauf"

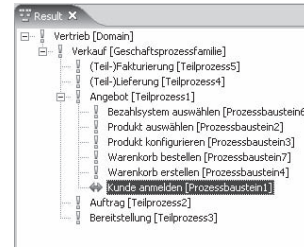


Abb. 5: Merkmalmodell des kundenspezifischen GP "Verkauf"

Die Transformation des kundenspezifischen GP soll mit Hilfe eines Generators realisiert werden. Im Beispiel ist es ein Generator für die Zielsprache BPEL. Als Endergebnis erhält man ein einsatzfähiges BPEL-Projekt „GP\_Verkauf“ für das Anwendungssystem.

### 2.3 Architektur

Die Process Factory stellt Transformationsprozesse bereit, die aus domänenindividuellen Informationen und Anforderungen konfigurierbare Modelle in Form von Templates von GPF für Referenz-Organisations- oder Referenz-Anwendungssystem-Modelle entwickeln. Domänenindividuelle Informationen stellen Wissen und die domänenindividuellen Anforderungen beschreiben Regeln bzw. Zusammenhänge der betrachteten Domäne (z.B. Vertrieb). Die GPF-Entwicklung (Abb. 6) wird in die Phasen Analyse, Konstruktion und Bereitstellung gegliedert, in denen verschiedene Techniken zur Verfügung stehen.

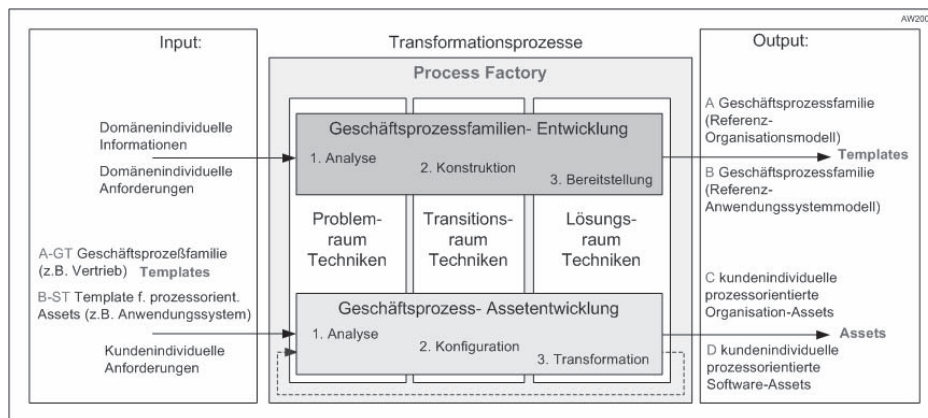


Abb. 6: Process Factory Architecture

Problemraum-Techniken stellen Technologien zur Analyse und Beschreibung der zu untersuchenden Domäne bereit. Transitionsraum-Techniken beherbergen eine Menge von Modellierungs- und Konfigurations-Techniken zum Aufbau und Instanziierung von Modellen. Techniken zur datenbank-gestützten Bereitstellung sowie zur generativen Umwandlung von Modell-Artefakten sind im Lösungsraum definiert. Im zweiten Entwicklungsvorgehen (Abb. 6), GP-Assetentwicklung genannt, werden aus domänen-spezifischen Templates der GPF konkrete kundenindividuelle prozessorientierte Assets entwickelt.

### 3 Diskussion und Ausblick

Erste praxisnahe Untersuchungen wurden bei einem Systemhaus für ERP-Software durchgeführt [St04]. In weiteren Arbeiten erfolgt die jeweils separate Beschreibung der zwei Entwicklungsphasen einer Process Factory, die GPF-Entwicklung und die GP-Assetentwicklung. Wobei jeweils schwerpunktmäßig auf die zweite Phase (Konstruktion, Konfiguration) eingegangen wird. Eine genauere Untersuchung der Beziehungen zwischen Process Factory und Software Factory ist für die Realisierung der Idee eines Mass Customization von Anwendungssystemen von grundlegender Bedeutung. Die Untersuchung der Anforderungen an ein softwaregestütztes Werkzeug zur Realisierung einer Process Factory stehen noch an, besonders in Bezug zur Konstruktion von GPF. Hierzu werden zwei Forschungsprojekte mit sächsischen KMUs vorbereitet.

### Literaturverzeichnis

- [Aene] Produkt Aeneis Version 5.0 von ATOSS AG. <http://www.aeneis.de>.
- [BPEL] Business Process Execution Language for Web Services Version 1.1. <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbizspec/html/bpel1-1.asp>, abgerufen am 17.01.2005.
- [GS04] Greenfield, J., Short, K. Software Factories: Assembling Applications with Patterns, Modells, Frameworks, and Tools. Wiley&Sons, Indianapolis 2004.
- [Kr97] Körmeier, K. Zielorientierte Festlegung von Geschäftsprozessen. Diss., Dresden 1997.
- [La97] Lang, K. Gestaltung von Geschäftsprozessen mit Referenzprozeßbausteinen. Gabler, Wiesbaden 1997.
- [Pi94] Pine, B., J. Massgeschneiderte Massenfertigung: neue Dimensionen im Wettbewerb. Wirtschaftsverl. Ueberreuter, Wien 1994.
- [Re97] Remme, M. Konstruktion von Geschäftsprozessen. Gabler, Wiesbaden 1997.
- [SAPN] Produkt SAP Netweaver von SAP AG. <http://www.sap.de/netweaver>.
- [Sc98] Scheer, A.-W. ARIS – vom Geschäftsprozeß zum Anwendungssystem. Springer, Berlin et al. 1998.
- [Sp04] Speck, A. Variabilitäten in dynamischen Abläufen: Modellierung und Validierung. Habil.-Schrift, Jena 2004.
- [St04] Steinert, M. Modellierung von Geschäftsprozessfamilien mit der UML 2.0. Diplomarbeit, Universität Leipzig, 2004.
- [We05] Werner, A. Kontext und Architektur einer Process Factory. Unveröffentlichter Artikel, Universität Leipzig, 2005.