

Flexibilität für PLM-Services in der Digitalen Fabrik

Karl Flieder

Externer Lektor für Geschäftsprozessmanagement sowie für
Business Process Management & Engineering (BPME)
Academia Nova, Schwechat
eai@karlflieder.at

Abstract: Eine durchgängige digitale Abbildung von Geschäftsprozessen, die sich in der Fertigungsindustrie von der Produktentstehung über die Fertigung bis hin zu Serviceprozessen erstrecken kann, unterstützt die geschäftliche Flexibilität und damit auch die Zukunftsfähigkeit von Anwendungen ganz wesentlich. Durch eine Integration geometrischer Daten und Informationen, wie sie im Rahmen der Digitalen Fabrik und auf der Basis einer serviceorientierten Architektur (SOA) Anwendung finden, können im Engineering Entwicklungs- und Änderungszyklen verkürzt, Prozesse stabilisiert sowie Qualität, Zeit und Kosten optimiert werden. Dieser Praxisbeitrag beantwortet die Frage wie Services für das Product Lifecycle Management (PLM) für mehrere Produktlinien wiederverwendbar gestaltet werden können.

1 Einleitung

Der Begriff Geschäftsprozessmanagement umfasst im Sinne einer allgemeinen Managementdefinition nicht nur die strategische und operative Planung von Prozessen, sondern auch deren Implementierung, Ausführung und Kontrolle [BR07]. Aus technischer Sicht wird damit zumeist eine Assoziation zu bestehenden Technologien der verbindenden oder verknüpfenden Prozessintegration wie Enterprise Application Integration (EAI) und serviceorientierten Architekturen (SOA) hergestellt ([AW09], [FNS06], [He07]). Eine serviceorientierte Architektur steht dabei für ein aktuelles Paradigma, ein Denkmuster, zur Gestaltung von Unternehmensarchitekturen [SA08], damit auch zur Steigerung der Flexibilität, die verschiedene Facetten annehmen kann [HR+10]. Mit einer SOA wird das Ziel einer an die Geschäftsprozesse ausgerichteten Infrastruktur verfolgt, die flexibel und leicht adaptierbar sein soll. Im Rahmen der Implementierung von determinierten, kurz laufenden und datenzentrierten Prozessen kommen typischerweise XML-Technologien, Web Services und BPEL (Business Process Execution Language) auf breiter Ebene zum Einsatz [Fl10]. Fachliche Funktionen werden in Form von lose gekoppelten, dokumentierten und mehrfach nutzbaren Services bereitgestellt. Parallel zu diesem technikzentrierten Ansatz wird eine SOA aber auch als Managementkonzept verstanden. Aus Managementsicht wird mit einer SOA das Ziel einer an den Prozessen ausgerichteten Service-Landschaft verfolgt,

mit deren Hilfe agil und flexibel auf veränderte Anforderungen im geschäftlichen Umfeld reagiert werden soll. Während bei monolithischen Anwendungen (ERP-Systemen), die durch eine vereinigende Integration entstehen, die geschäftliche Funktionalität und die prozessuale Ablauflogik kaum getrennt voneinander betrachtet werden können, ist dies bei der verknüpfenden Integration auf SOA-Basis sehr wohl möglich. Damit können sowohl die Flexibilität als auch die Zukunftsfähigkeit einer Anwendung positiv beeinflusst werden. Die geschäftliche Funktionalität wird in elementaren Services abgebildet, die Implementierung der Prozess- oder Ablauflogik erfolgt durch eine Orchestrierung elementarer Services sowie durch eine variable Gestaltung der Wegwahl auf verschiedenen Ebenen (Prozesse, Module, Dokumente) und mit Hilfe von Variablen. In den folgenden Abschnitten wird ein Lösungsansatz zur flexiblen und modularen Gestaltung einer Integrationslösung im Rahmen des Product Lifecycle Managements (PLM) auf Basis einer SOA skizziert. Die Erhöhung der Prozessflexibilität wurde durch einen besonderen Gestaltungsansatz erreicht.

2 Variabilität durch Servicekontext

Aus technischer Sicht wird mit einer serviceorientierten Integrationsarchitektur das Ziel verfolgt, elementare fachliche Services zu schaffen, um durch eine Kombination dieser Services – je nach Bedarf – Prozesse flexibel gestalten zu können ([AG08], [La06]). Services sind damit die zentralen Elemente einer Integrationsarchitektur, die nach den Prinzipien einer SOA konzipiert wurde. Flexibilität gilt als einer der wichtigsten SOA-Architekturziele, die Bildung von Varianten gilt als eine anerkannte Möglichkeit zur Erhöhung der Flexibilität [AD08]. Ein elementarer Service implementiert in der Regel eine klar abgegrenzte geschäftliche Funktionalität und beinhaltet das gemeinsame Vielfache an „variablen Eigenschaften“ jener Produktlinien, für die er mehrfach verwendet werden soll. Im Rahmen einer SOA erfolgt die Zusammenstellung eines komplexen Dienstes, einer so genannten Composite Application (Abbildung 1), durch eine Orchestrierung elementarer Dienste. Im konkreten Fall wurde diese derart gestaltet, dass sie für verschiedene geschäftliche Projekte nutzbar wird. In ihrer Außenwirkung beruhen Services auf dem Konzept des Versteckens der eigenen Komplexität und geben ihrem Umfeld nicht preis, wie ihre Funktionalität implementiert wurde. In ihrem Inneren jedoch können sie mit einem Kontext versehen werden, der zur Laufzeit gezielt für eine Wiederverwendung genutzt werden kann. Ein nach außen hin statusloser Service gilt in der Regel als agnostisch im Sinne von „nicht wahrnehmend“ bzw. „ohne Wissen“ [Er08]. Innerhalb dieses Service jedoch kann gezielt eine nicht-agnostische Logik eingesetzt werden, um damit den Daten- und Nachrichtenfluss zur Laufzeit variabel steuern zu können. In der Integrationspraxis eines Fahrzeugherstellers ruft zum Beispiel ein Top-Level-Service, dieser erfüllt vornehmlich Steuerungsaufgaben, weitere Services auf, in denen fachliche Funktionalität und Verarbeitungslogik abgebildet wurde. Dabei verarbeitet die Composite Application Logik, Funktionalität und Daten aus verschiedenen Quellsystemen. Durch die Orchestrierung entstehen prozessuale Abläufe, die heterogene Systemgrenzen überwinden und flexible Anwendungen schaffen. Variablen und Parameter spielen dabei eine wichtige Rolle, um der Forderung nach Flexibilität und Agilität nachkommen zu können (vgl. [AT06]).

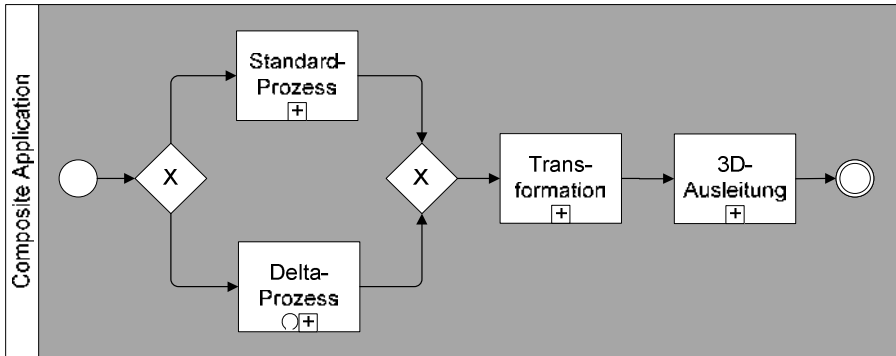


Abbildung 1: Schema einer PLM-Anwendung aus elementaren Services

Abbildung 1 zeigt die Orchestrierung einer PLM-Anwendung aus elementaren Diensten, die für verschiedene Produkte wiederverwendbar ist. Dies wird unter anderem durch eine generische Ablauflogik erreicht. Lediglich das variable Beziehungswissen zwischen den einzelnen Services, welches der Composite Application in Form von Parametern mitgegeben wird, ist für die jeweiligen Projekte unterschiedlich. Durch das über Prozessvariablen hergestellte Beziehungsgeflecht konnte eine flexible Prozesslogik für unterschiedliche fachliche Anforderungen etabliert werden. Die Prozess- bzw. Ablauflogik wird zur Laufzeit von Variablen beschrieben, die an jedem Knotenpunkt über das für die weitere Ausführung benötigte „Wissen“ verfügen. Bei Bedarf werden die variablen Anteile dieser Prozesslogik an die elementaren Services weitergereicht. Dadurch erhalten die nach außen hin statuslosen Services, in ihrem Inneren einen Kontext, mit dem zur Laufzeit der aktuelle Zustand in jedem Knotenpunkt beschrieben werden kann. Dieser Servicekontext wird für die Gestaltung der fachlichen Variabilität für mehrere Produktlinien benötigt. Der Service verhält sich zudem idempotent und ist daher mehrfach aufrufbar, ohne dass sich das Ergebnis der Verarbeitung ändert.

3 Anwendungsfall Digital Mock Up (DMU)

Unter dem Begriff „Digitale Fabrik“ versteht man das virtuelle Abbild einer geplanten oder realen Fabrik mit seinen Gebäuden, Einrichtungen, Anlagen und Fördersystemen sowie deren Wertschöpfungsprozessen in digitaler Form. Computer Aided Engineering (CAE), das die computerunterstützten Ingenieurleistungen Konstruktion und Berechnung umfasst, ist ein typischer Anwendungsbereich im Rahmen der Digitalen Fabrik. Für die Simulation von Engineering-Prozessen werden in Simulationsprogrammen zunächst Daten über die Produktionsabläufe und die zugehörige Logistik hinterlegt. Auf diese Weise können im Vorfeld der physischen Fahrzeugkonstruktion komplette Prozesse, wie zum Beispiel die automatische Geometrieprüfung, mittels Simulation untersucht und optimiert werden. Dabei kommen mathematische Modelle zur Darstellung der Prozesse und Abläufe ebenso zum Einsatz, wie Softwarewerkzeuge zur geometrischen Darstellung. Bei der DMU-Integration (Digital Mock Up), dem so genannten "digitalen Prototypenbau", geht es in erster Linie um die digitale Montage, um Untersuchungen einfacher Bauteilgruppen sowie um die Beherrschung komplexer Produktstrukturen

außerhalb des CAD-Systems (Computer Aided Design) [HHD08]. DMU ist ein Überbegriff für verschiedene Softwarebausteine, die in verschiedenen Phasen des Entwicklungsprozesses eingesetzt werden, um frühzeitig Kontrollen zur Beseitigung von Fehlern und zur Einsparung von Zeit und Kosten durchführen zu können. Diese Untersuchungen erfolgen auf der Basis von 3D-CAD-Modellen und umfassen die Absicherung der geometrischen Stimmigkeit aller Bauteile in verschiedenen Fahrzeugkonfigurationen sowie die Berechnung von Kollisions-, Kontakt- und Abstandsproblemen. Ein wesentliches wirtschaftliches Ziel dieser Simulation ist eine möglichst frühe Kenntnis allfälliger Fehlersituationen, um damit Zeit und Kosten sparen sowie die Qualität erhöhen zu können. Dabei werden mechanische Kräfte simuliert, unterschiedliche Varianten berechnet und Ergebnisse beurteilt [DH09]. Im Zuge der gegenständlichen SOA-Implementierung wurde darüber hinaus das methodische Ziel verfolgt, die Services für verschiedene Produktlinien wiederverwendbar zu gestalten. Dieses Ziel wurde durch den konsequenten Einsatz von Variablen und Parametern auf verschiedenen Abstraktionsebenen erreicht: *Prozessebene* (Service-Orchestrierung), *Modulebene* (fachliche Logik) und *Dokumentebene* (Austauschformat) [FI09].

3.1 Prozessebene

Auf der Prozessebene wird das Beziehungswissen zwischen den einzelnen Services durch Prozessvariablen hergestellt. Der eingesetzte Enterprise Service Bus kann einzelne Werte, aber auch komplette XML-Dokumente, die zum Beispiel die Beschreibung von einzelnen Teilen, Komponenten oder Baugruppen beinhalten, in Variablen speichern. Deren Inhalte werden zur Laufzeit an die elementaren Services weitergereicht und damit für die Ablaufsteuerung nutzbar gemacht. Die Orchestrierung der Services zu einer Composite Application erfolgt über HTTP-Konnektoren, da keine dynamische Adressierung, wie sie mit BPEL-Controls und Web Service Schnittstellen möglich wäre, notwendig ist. Dabei können wahlweise globale Header-Variablen – zum Beispiel mit dem Programm `curl` – und/oder lokale Prozessvariablen verwendet werden. Die Interaktion zwischen den einzelnen Services erfolgt auf der Basis von Nachrichten. Diese rufen entfernte Funktionen auf und transportieren XML-Dokumente mit fixen Inhalten ebenso wie Ein- und Ausgangsparameter mit variablen Inhalten. Die Auswertung der vom Fachbereich vorgegebenen Konfigurationsdaten erfolgt auf der Basis von Business Rules [FI10]. Auf diese Weise entsteht eine für verschiedene Produkte verwendbare generische Auflauflogik.

3.2 Modulebene

Die einzelnen Module eines Service, die zum Beispiel für die Transformation und die hierarchische Strukturierung von XML-Dokumenten genutzt werden, enthalten Funktionalität, die durch XSLT (eXtended Stylesheet Language) und XPath (XML Path Language) bereitgestellt wird. Lokale und globale Variablen bilden dabei wiederum das Beziehungswissen zwischen den einzelnen Templates innerhalb der Module ab. Dies geschieht in der Weise, dass die Eingabeparameter von XSLT-Modulen wie Parameter einer Klassenfunktion behandelt werden. Eine Variable stellt damit einen leeren „Container“ für fachliche Inhalte bereit, der je nach Projekt mit unterschiedlichen

Werten gefüllt wird. Für die Konvertierung einer flachen XML-Struktur, in ein in seiner Tiefe unbegrenztes hierarchisches XML-Dokument, wird das notwendige Beziehungswissen zwischen den einzelnen Baugruppen, Komponenten, Einzelteilen und deren Attributen während der Verarbeitung ebenfalls in Form von Parametern und Variablen bereitgestellt.

3.3 Dokumentebene

Ein Fahrzeug besteht auch in seiner digitalen Repräsentation aus verschiedenen Baugruppen, eine Baugruppe besteht aus mehreren Komponenten, die sich wiederum aus einzelnen Teilen zusammensetzen und durch das Geometriemodell dreidimensional (3D) beschrieben werden. Ein PDM-System (Produktdatenmanagement) bildet all diese Beziehungen und geometrischen Informationen ab und stellt sie in Form eines XML-Dokuments für den Datenaustausch bereit. Das Schema PLM-XML ist ein Industriestandard für den PLM-Datenaustausch [AMR10], das die Wiederverwendung gleicher Konstrukte für verschiedene Projekte unterstützt. Dabei verwenden zwei oder mehrere Partner ein und dasselbe Schema, um den Inhalt (Syntax) und die Bedeutung (Semantik) der Daten auf die gleiche Art und Weise interpretieren zu können. In einem „flachen“ PLM-XML-Dokument, werden die Zusammenhänge zwischen den Attributen, z.B. Product (Sachnummer), ProductView (Struktur), ProductRevision (Änderungsstand), RevisionRule, ReleaseStatus, DataSet und ExternalFile (Pfadangaben) über variable, verbindende Attribute hergestellt. Die Verknüpfung dieser Informationen, wie sie für die Erstellung einer hierarchischen Baumstruktur benötigt werden, übernimmt der SOA-Entwickler nach fachlichen Vorgaben mit Hilfe von XSLT und XPath unter Nutzung der deskriptiven Eigenschaften der genannten XML-Technologien.

4 Zusammenfassung

In diesem Beitrag wurde ein SOA-basierter Lösungsansatz für eine PLM-Integration vorgestellt, bei dem eine wiederverwendbare, generische Ablauflogik im Vordergrund stand. Durch eine konsequente Berücksichtigung von variablem Beziehungswissen über verschiedene Abstraktionsebenen hinweg konnte eine Wiederverwendung für mehrere Produktlinien erreicht werden. Die Orchestrierung der elementaren Services zu einer Composite Application erfolgte mit Konnektoren, die das universelle Protokoll HTTP (Hypertext Transfer Protocol) nutzen. Im Zusammenwirken mit dem verwendeten Enterprise Service Bus (ESB) eröffnete sich dabei die Möglichkeit neben den lokalen Variablen innerhalb der Services auch die globalen Header-Variablen der HTTP-Aufrufe über die gesamte Composite Application hinweg zu nutzen. Weitere Eckpunkte dieser Integrationslösung sind die Nutzung anerkannter Methoden wie Enterprise Integration Patterns (z.B. Canonical Data Format und Event-driven Consumer) [HW04] sowie von domänenspezifischen Austauschformaten wie PLM-XML. Bedingt durch die hohe Anzahl an Bauteilen, die in den XML-Dokumenten hierarchisch abgebildet werden mussten, war bei der Implementierung zudem auf ein günstiges Laufzeitverhalten Bedacht zu nehmen.

Literaturverzeichnis

- [Ad08] Adam, S.; Doerr, J.: How to better align BPM & SOA – Ideas on improving the transition between process design and deployment. In Proceedings of BPMDS 2008, <http://lams.epfl.ch/conference/bpmds08/program/paper6.pdf> (abgerufen am 29.06.2010)
- [AMR10] Aderl, R.; Mosch, C.; Rollmann, T.: Neutraler Datenaustausch in der Digitalen Fabrik. In ZWF 105 (2010) 3, Carl Hanser Verlag, München 2010, S. 163-167.
- [AG08] Aier, S.; Gleichauf, B.: Begründung eines differenzierten Serviceverständnisses und Richtlinien für die Servicekonstruktion. In Proceedings 3. Workshop BSOA 2008, Shaker Verlag, Leinfelden b. Stuttgart 2008, S. 47-58.
- [AT06] Ackermann, J.; Turowski, K.: Zur Rolle von Parametrisierung bei der fachlichen Anpassung betrieblicher Softwarekomponenten. In Schelp, J.; Winter, R.; Frank, U.; Rieger, B.; Turowski, K. (Hrsg.): Integration, Informationslogistik und Architektur. Proceedings DW2006, Gesellschaft für Informatik, Bonn 2006, S.341-359.
- [AW09] Aier, S.; Winter, R.: Virtuelle Entkopplung von fachlichen und IT-Strukturen für das IT/Business-Alignment: Grundlagen, Architekturgestaltung und Umsetzung am Beispiel der Domänenbildung. In Wirtschaftsinformatik 51 (2009) 2, S. 175-191.
- [BR07] Brabänder, E.; Erbach, F.: “AVE for SAP” – Eine Vorgangsweise für das Geschäftsprozessmanagement mit SAP. In Informatik Spektrum 30 (6) 2007, S. 419-427.
- [DH09] Dietrich, W.; Heilig R.: Effiziente Kollisionsprüfung in der Fahrzeugtechnik. In Digital Engineering Magazin 1/2009, S. 72-74.
- [Er08] Erl, T.: Introducing SOA Design Patterns. SOA World Magazin, Vol. 9, 6/2008.
- [Fl09] Flieder, K.: Serviceorientierte Softwarearchitekturen in Theorie und Praxis. In e&i Elektrotechnik & Informationstechnik 126 (12) 2009, Springer-Verlag, Berlin, Heidelberg, New York 2009, S. a32-a35.
- [Fl10] Flieder, K.: Mit RFID und BPM zum ereignisgesteuerten Unternehmen. In ZWF 105 (2010) 5, Carl Hanser Verlag, München 2010, S. 494-502.
- [FNS06] Fischer, D.; Nirsberger, I.; Stelzer, D.: Ein Modell zur Bestimmung des Grades der unternehmensübergreifenden Integration von Informationssystemen. In Schelp, J.; Winter, R.; Frank, U.; Rieger, B.; Turowski, K. (Hrsg.): Integration, Informationslogistik und Architektur. Proceedings DW2006, Gesellschaft für Informatik, Bonn 2006, S. 427-447.
- [He07] Heutschi, R.: Serviceorientierte Architektur: Architekturprinzipien und Umsetzung in der Praxis. Springer-Verlag, Berlin, Heidelberg, New York 2007.
- [HHD08] Hirz, M.; Hirschberg, W.; Dietrich, W.: Integrated 3D-CAD Design Methods in Early Automotive Development Processes. In Proceedings FISITA 2008 – The Future of Automobiles and Mobility, S. 1-15.
- [HR+10] Holschke, O.; Rake, J.; Offermann, P.; Bub, U.: Steigerung der Softwareflexibilität bei Geschäftsprozessänderungen. In Wirtschaftsinformatik 52 (2010) 1, S. 3-15.
- [HW04] Hohpe, G.; Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Springer-Verlag, Berlin, Heidelberg, New York 2007.
- [La06] Laures, G.: Flexibilitätsanalyse service-orientierter Architekturen zur Realisierung von Geschäftsprozessen. In Proceedings EMISA 2006, Springer-Verlag, Berlin, Heidelberg, New York 2006, S. 163-177.
- [SA08] Stutz, M.; Aier, S.: Vorgehensmodell zur fachlichen Bewertung serviceorientierter Architekturen. In Bichler, M.; Hess, T.; Krcmar, H.; Lechner, U.; Matthes, F.; Picot, A.; Speitkamp, B.; Wolf, P. (Hrsg.): Proceedings Multikonferenz Wirtschaftsinformatik 2008, GITO-Verlag, Berlin 2008, S. 989-999.