

Ein Metamodell zur integrierten Darstellung von aktuellen Konzepten zur Industrialisierung der Softwareentwicklung

Oliver Höß, Jens Drawehn, Anette Weisbecker

Competence Center Software-Management
Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO)
Nobelstraße 12
70569 Stuttgart

{Oliver.Hoess | Jens.Drawehn | Anette.Weisbecker}@iao.fraunhofer.de

Abstract: Durch die Notwendigkeit der Steigerung von Effizienz und Qualität in der Softwareentwicklung wurden in den letzten Jahren eine Reihe von Konzepten für die Industrialisierung der Softwareentwicklungsprozesse entwickelt. Beispiele hierfür aus dem eher technologisch orientierten Bereich sind Serviceorientierte Architekturen (SOA), Business Process Management (BPM), komponentenbasierte und modellbasierte Softwareentwicklung sowie Software-Produktlinien. Trotz einer Vielzahl von Arbeiten besteht in diesen Bereichen oftmals Unklarheit über die Bedeutung von einzelnen Begriffen sowie insbesondere über die Zusammenhänge zwischen diesen Konzepten. Der vorliegende Beitrag beschreibt einen Ansatz eines Metamodells für die integrierte Darstellung dieser Konzepte, in dem die wesentlichen Begriffe sowie die Beziehungen zwischen den Begriffen erläutert werden. Der Beitrag ist als Diskussionsgrundlage zu verstehen, um die Entwicklung einer Domänen-Ontologie für aktuelle Konzepte in Software-Management und -Technik voranzutreiben.

1 Einleitung

Die Erkenntnis, dass Software ein wesentlicher Bestandteil der Wertschöpfungskette der Unternehmen und somit ein zentraler Erfolgsfaktor für deren Wettbewerbsfähigkeit ist, hat sich bereits seit einiger Zeit durchgesetzt [Sp04].

Daraus folgt, dass eine Steigerung der Effizienz und Qualität der Softwareentwicklung und damit der resultierenden Software einen wesentlichen Beitrag zur Steigerung der Wettbewerbsfähigkeit der Unternehmen leisten kann. Ansätze hierfür sind dabei u. a. in der Industrialisierung der Softwareentwicklung durch den ingenieurmäßigen Einsatz von aktuellen Methoden, Technologien und Werkzeugen zu sehen [Ro03].

Beispiele von aktuellen Ansätzen, die ein großes Potenzial versprechen, sind Serviceorientierte Architekturen (SOA) und Business Process Management (BPM). Ein wenig länger sind bereits die Konzepte der komponentenbasierten Softwareentwicklung, der modellbasierten Softwareentwicklung und Software-Produktlinien im Gespräch.

Leider ist es im IT-Umfeld üblich, dass der Markt sehr von Schlagworten getrieben wird und Marketing-Interessen oft vor Inhalte gestellt werden. In Kombination mit fehlenden bzw. allgemein akzeptierten Definitionen (z. B. für SOA) werden dadurch oft Missverständnisse hervorgerufen und es werden unnötige Diskussionen geführt, z. B. bezüglich der trennenden Abgrenzung der genannten Konzepte. Aus Sicht der Autoren harmonisieren jedoch die oben genannten Konzepte sehr gut und es besteht eine Vielzahl von Zusammenhängen.

Ziel des vorliegenden Beitrags ist es daher, die Konzepte SOA, BPM, komponentenbasierte Softwareentwicklung, modellbasierte Softwareentwicklung und Software-Produktlinien in einen Kontext zu bringen. Dabei wird ein ontologie-basierter Ansatz gewählt, d. h. es wird ein Metamodell definiert, bei dem die jeweiligen Entitäten über unterschiedliche Beziehungen miteinander verknüpft sind. Der Fokus dieses Beitrags liegt weniger in der Tiefe der Betrachtung der einzelnen Themenfelder, sondern in der Darstellung von Beziehungen zwischen den Begriffen aus unterschiedlichen Konzepten.

Nach dieser Einleitung werden in Kap. 2 existierende Arbeiten in diesem Umfeld ohne Anspruch auf Vollständigkeit kurz dargestellt. In Kap. 3 wird eine grobe Definition der Begriffe und Beziehungen im Themenfeld SOA als zentrales Konzept dieses Beitrags durchgeführt. In Kap. 4 wird dieses Basismodell schrittweise um Begriffe aus den Bereichen BPM, komponentenbasierte Softwareentwicklung, modellbasierte Softwareentwicklung und Software-Produktlinien ergänzt. Kap. 5 fasst den Beitrag zusammen und gibt einen Ausblick auf zukünftige Entwicklungen.

2 Existierende Arbeiten

In den für diesen Beitrag relevanten Bereichen wurden für sich genommen jeweils eine Vielzahl von Arbeiten durchgeführt, die die gesamte Palette von der eher grundlagenorientierten Forschung bis hin zur Anwendung in der Praxis abdecken [Hö07b; SW08; We02; SWH05; LSR07].

Im Bereich der Serviceorientierten Architekturen (SOA) hat die herstellerunabhängige Organisation OASIS (Organization for the Advancement of Structured Information Standards) ein SOA-Referenzmodell bzw. ein Metamodell veröffentlicht, das prinzipiell den Bereich SOA abdeckt, aber weder in der Tiefe und Qualität zufriedenstellend ist noch angrenzende Bereiche und Konzepte betrachtet [Oa06].

Auch in [En08] werden an mehreren Stellen Begriffsnetze im Kontext von Serviceorientierten Architekturen definiert, die auch die fachlichen Aspekte im IT-Architektur-Management berücksichtigen. Noch einen Schritt weiter gehen heute bereits Unternehmen, wie z. B. die EnBW (Energie Baden-Württemberg), die bereits ein sehr umfangreiches SOA-Metamodell entwickelt hat, das alle Aspekte von der serviceorientierten Analyse, über das Design bis hin zu Implementierung und Betrieb einer SOA inkl. der dazugehörigen Werkzeuge und Rollen abdeckt [SS07].

Auch im Rahmen der Gesellschaft für Informatik gibt es Bestrebungen, Begriffsnetze zu Themen der Softwaretechnik aufzubauen. Im Rahmen des Aufbaus des Informatik-Begriffsnetzes¹ wurden bereits drei Begriffssammlungen zu den Themen Analyse und Modellierung von Anwendungssystemen, Objektorientierte Modellierung und Vorgehensmodelle entwickelt [Bi07]. Zu dem aktuellen Thema SOA konnten an dieser Stelle jedoch keine Informationen und Erläuterungen gefunden werden.

3 Serviceorientierte Architekturen

Serviceorientierte Architekturen (SOA) sind ein derzeit sowohl in der Wissenschaft als auch in der Praxis stark diskutiertes Thema und versprechen im Gegensatz zu gewachsenen IT-Systemen und IT-Landschaften mehr Flexibilität und Effizienz sowie eine bessere Unterstützung von sich ändernden Geschäftsprozessen [Hö07a; Jo08].

Die Grundideen im Umfeld der Serviceorientierten Architekturen sind nicht neu, dennoch haben sich in den letzten Jahren durch offene Standards aus dem Web-Service-Umfeld sowie aus dem Workflow-Umfeld Möglichkeiten zur standard-konformen Realisierung einer SOA ergeben.

Im Folgenden sollen die wesentlichen und im Kontext dieses Beitrags relevanten Konzepte einer SOA kurz dargestellt werden (siehe auch Abbildung 1).

An zentraler Stelle steht das Konzept des *Service*, der eine wohldefinierte Funktionalität über standardisierte Schnittstellen (z. B. SOAP²) zur Verfügung stellt. Ein Service kann ein Basis-Service (*Basic Service*) sein, der relativ einfache Funktionalitäten bereit stellt oder ein zusammengesetzter Service (*Composite Service*), der die Funktionalitäten mehrerer *Basic Services* oder *Composite Services* nutzt und daher aus diesen aggregiert wird.

Ein wesentlicher Vorteil einer SOA in der vollen Ausbaustufe ist, dass die einzelnen *Services* nicht manuell, z. B. mit Java oder .NET-Code, zu *Composite Services* aggregiert werden. Stattdessen wird ein *IT-Prozessmodell*, das den zu realisierenden *IT-Prozess* beschreibt, direkt genutzt, um einzelne *Services* zu höherwertigen *Composite Services* zu aggregieren. Dieser Prozess wird Orchestrierung genannt und wird in den derzeit am Markt verfügbaren Produkten vor allem durch den Einsatz des Standards BPEL (Business Process Execution Language³) unterstützt. Dadurch kann eine prozessgesteuerte IT realisiert werden (siehe auch 4.1).

¹ <http://public.tfh-berlin.de/~giak/>

² ursprünglich Simple Object Access Protocol (<http://www.w3.org/TR/soap/>)

³ http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

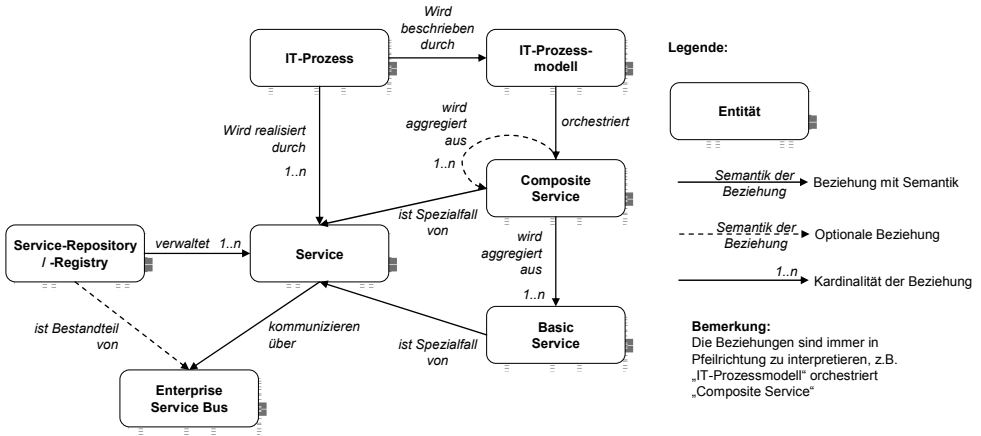


Abbildung 1: Grundkonzepte im SOA-Umfeld

Die Services werden in einem *Service-Repository* bzw. einer *Service-Registry* verwaltet. Oftmals werden *Registry* und *Repository* dadurch unterschieden, dass in der *Registry* vor allem technische Informationen (z. B. Schnittstellen) gespeichert sind, die zur Laufzeit benötigt werden und im *Repository* zusätzliche Meta-Daten bzgl. den einzelnen Services. Im Kontext dieses Beitrags werden die beiden Konzepte jedoch zusammengezogen, was in der Praxis auch oft durch dadurch abgebildet wird, dass sie durch identische Produkte abgebildet werden.

Die *Services* kommunizieren über einen sog. *Enterprise Service Bus (ESB)*, der eine einheitliche Kommunikations- und Sicherheits-Infrastruktur sowie Adapter und Transformationsfunktionalitäten beinhaltet. Ein ESB bildet das Rückgrat einer SOA und muss daher auf Performanz und Ausfallsicherheit ausgelegt sein. Oftmals ist auch ein *Repository* bzw. eine *Registry* ein Bestandteil des *Enterprise Service Bus*.

4 Integration weiterer Konzepte

Nachdem in Kap. 3 die Grundkonzepte der Serviceorientierten Architekturen vorgestellt wurden, wird in diesem Kapitel die Integration der Konzepte Business Process Management, komponentenbasierte Softwareentwicklung, modellbasierte Softwareentwicklung und Software-Produktlinien beschrieben. Der Fokus liegt dabei weniger auf einer Tiefe der Betrachtungen sondern in der Darstellung der wesentlichen Berührungspunkte.

4.1 Business Process Management

Business Process Management (BPM), d. h. das Management von Geschäftsprozessen, ist bereits seit einigen Jahren ein wichtiges Thema bei Unternehmen. Dabei ist sowohl die Analyse als auch die Modellierung und anschließende Optimierung von Geschäftsprozessen von Relevanz. Eine aktuelle Marktübersicht über 18 BPM-Werkzeuge wurde durch das Fraunhofer IAO durchgeführt [SW08].

Derzeit gewinnt das Thema BPM im Kontext von SOA zusätzlich an Bedeutung. Der Fokus liegt dabei auf der Umsetzung der Geschäftsprozesse durch IT-Systeme, insbesondere durch die direkte Ausführung der modellierten Prozesse durch die IT-Systeme. Eine möglichst gute und effiziente Unterstützung der Geschäftsprozesse durch die IT-Systeme ist eines der wesentlichen Versprechen einer SOA [Hö07b].

Daher soll in diesem Abschnitt der Zusammenhang zwischen BPM und SOA vereinfacht dargestellt werden (siehe auch Abbildung 2). Oftmals werden derzeit diese Begriffe - teilweise auch aus Marketing-Gründen - vermischt, was nach Sicht der Autoren unnötig ist, da klar definierte Beziehungen bestehen. Dies wird auch durch den SOA-Check 2007 bestätigt, einer Umfrage, die von der Amadee AG und vom Wolfgang Martin Team durchgeführt wurde und in Form einer Studie veröffentlicht wurde [Am07]. Dort assoziiert nur eine Minderheit von 24% SOA mit der kompletten Restrukturierung von Unternehmensprozessen.

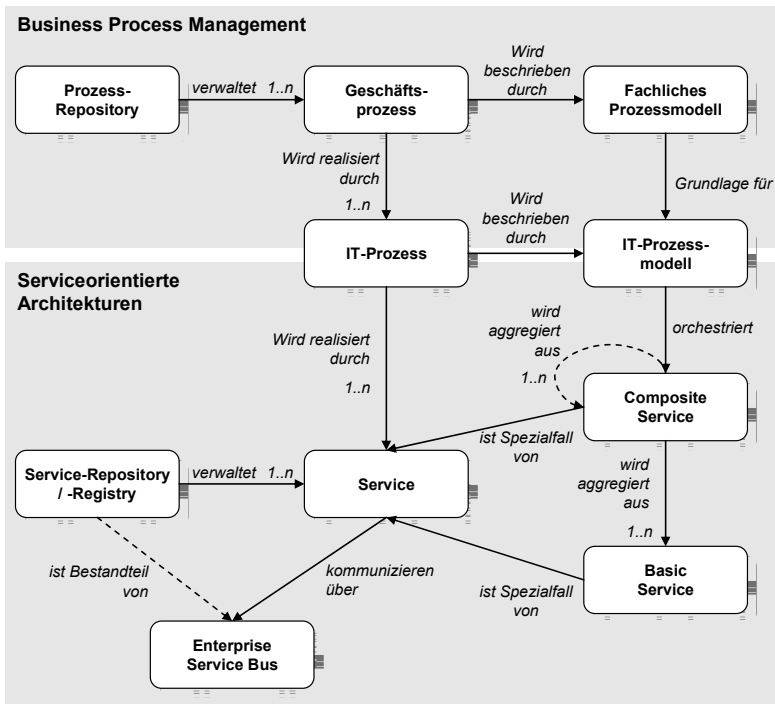


Abbildung 2: Integration von BPM-Konzepten

Die in Kap. 3 beschriebenen *IT-Prozesse* sowie die dazugehörigen *IT-Prozessmodelle*, die eine SOA, die die Orchestrierung im Rahmen einer SOA steuern, stehen in einem engen Zusammenhang mit den fachlichen *Geschäftsprozessen* sowie den dazugehörigen *fachlichen Prozessmodellen*, die diese beschreiben. Die *IT-Prozesse* setzen die *Geschäftsprozesse* in den IT-Systemen auf Basis einer SOA um. Die entsprechenden Prozessmodelle, d. h. das fachliche *Prozessmodell* sowie das *IT-Prozessmodell*, können voneinander abgeleitet werden.

Da zwischen den *fachlichen Prozessmodellen* und den *IT-Prozessmodellen* eine sog. „semantische Lücke“ besteht, ist eine automatische Ableitung problematisch. Im Forschungsumfeld arbeiten eine Reihe von Vorhaben an diesem Thema, z. B. das vom Bundesministerium für Bildung und Forschung geförderte Projekt OrViA [SI07].

Wird in einem Unternehmen wirklich ein systematisches Business Process Management umgesetzt, was derzeit leider nur bei einer Minderheit der Unternehmen der Fall ist, ist es notwendig, ein Werkzeug zu verwenden, mit dem die *Geschäftsprozesse* verwaltet werden, also ein *Prozess-Repository*. Dieses kann mit den heute am Markt verfügbaren BPM-Werkzeugen realisiert werden [SW08].

Insgesamt beschäftigt sich BPM also mit der Optimierung von *Geschäftsprozessen* und SOA mit der Optimierung der entsprechenden *IT-Prozesse* bzw. IT-Systeme. Beide Konzepte ergänzen sich jedoch sehr gut und besitzen eine enge inhaltliche Verbindung (siehe Abbildung 2).

4.2 Komponentenbasierte Softwareentwicklung

Vor ca. 10 Jahren war das Schlagwort der „komponentenbasierten Softwareentwicklung“ (component-based software development, CBSD) mit dem Aufkommen von Technologien wie CORBA (Common Object Request Broker Architecture), ActiveX oder EJBs (Enterprise Java Beans) in aller Munde. Das Versprechen war vor allem, durch die Wiederverwendung von bestehenden *Komponenten* eine Steigerung der Effizienz und Effektivität der Softwareentwicklung sowie eine gesteigerte Qualität der entstehenden Software-Produkte zu erreichen [We02]. Das Versprechen konnte auch in vielen Bereichen erfüllt werden. Insbesondere durch den extensiven Einsatz von OpenSource-Komponenten kann heutzutage Software teilweise sehr effizient und kostengünstig entwickelt werden [Hö05].

Oftmals ist jedoch der Zusammenhang zwischen dem Konzept der komponentenbasierten Softwareentwicklung und dem Konzept SOA unklar. Oftmals wird dargestellt, dass das Konzept der komponentenbasierten Softwareentwicklung durch das Konzept der serviceorientierten Architektur obsolet geworden ist.

Aus Sicht der Autoren ist diese Aussage jedoch nicht korrekt. Stattdessen kann ein sehr einfacher Zusammenhang hergestellt werden, da auch mit dem Aufkommen von SOA „traditionelle“ *Komponenten* benötigt werden, mit denen die *Services* bzw. *Basic Services* auf unterster Ebene realisiert werden (siehe Abbildung 3).

Es macht daher nach wie vor Sinn, die vorhandenen Komponenten in einem *Komponenten-Repository* zu verwalten. Dies kann unternehmensintern eingesetzt werden oder es können öffentliche im Internet verfügbare Repositories, z. B. für Open Source Komponenten, genutzt werden.

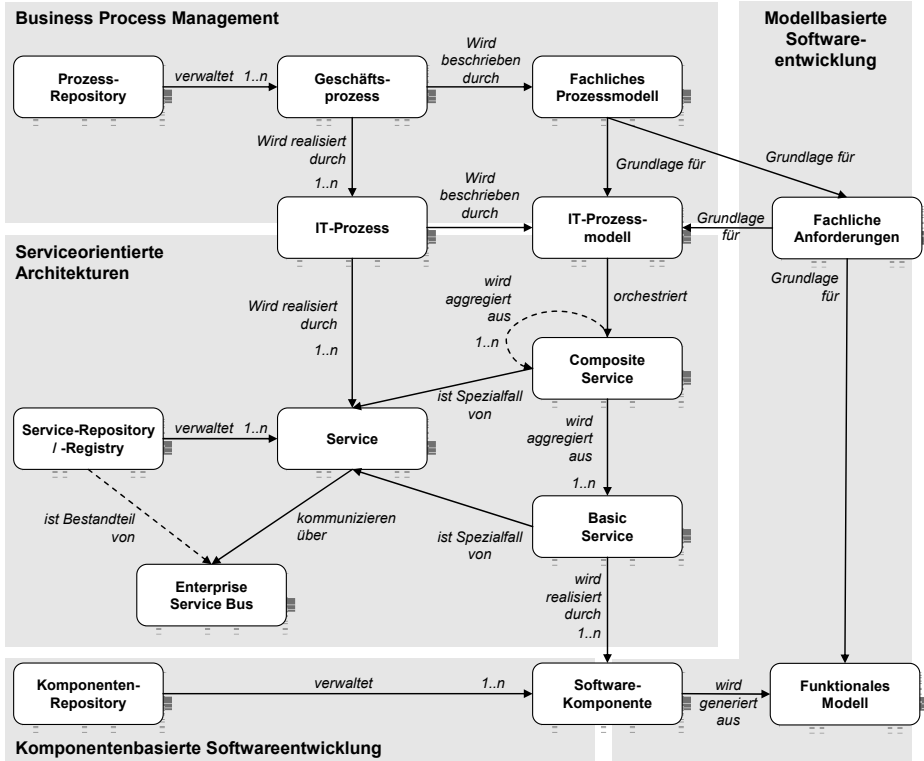


Abbildung 3: Komponentenbasierte und modellbasierte Softwareentwicklung

4.3 Modellbasierte Softwareentwicklung

Auch das Konzept der „modellbasierten Softwareentwicklung“ (model-based software development, MBSD) kann sehr gut in das beschriebene Metamodell integriert werden (siehe Abbildung 3). Die Grundidee dabei ist es, dass die Funktionalität von *Komponenten* nicht manuell in einer herkömmlichen Programmiersprache codiert wird, sondern stattdessen ein logisch abstrakteres *funktionales Modell* erstellt wird, aus dem die *Komponenten* generiert werden [SWH05].

Das *funktionale Modell* wird wiederum aus den *fachlichen Anforderungen* abgeleitet, die wiederum aus dem *fachlichen Prozessmodell* abgeleitet werden. Alternativ zu einer Generierung der *Komponenten* kann das *funktionale Modell* auch zur Konfiguration der *Komponenten* zur Laufzeit verwendet werden. Unter diesem Gesichtspunkt ist auch die Orchestrierung einer SOA durch *IT-Prozessmodelle* (z. B. in BPEL) als modellbasierte Softwareentwicklung zu verstehen (vgl. Kap. 3 und 4.1).

4.4 Software-Produktlinien

Die in 4.2 beschriebene „traditionelle“ Wiederverwendung auf Basis von Komponenten, die in einem Repository verwaltet werden, führt aufgrund einer Vielzahl von Problemfeldern, sowohl im organisatorischen als auch im technischen Umfeld, oftmals nicht zum Erfolg [Hö05].

Wenn man vor allem Systeme bzw. Software-Produkte in einer einzigen Domäne bzw. mit einem gemeinsamen Systemkern entwickelt, hat sich der Ansatz der Software-Produktlinien durchgesetzt, der bereits in einer Vielzahl von Anwendungsfällen erfolgreich eingesetzt wurde [LSR07].

Die über die verschiedenen Produkte identischen Funktionalitäten werden dabei in einer *Produktlinienplattform* zusammengefasst, die dann als gemeinsame Basis für die verschiedenen Produkte verwendet wird [PBL05]. Zu dieser gehören, wenn der Produktlinien-Ansatz eingesetzt wird, innerhalb des in diesem Artikel dargestellten Modells *Software-Komponenten*, *Basic* und *Composite Services* sowie *IT-Prozessmodelle*. Die Produktlinien-Plattform kann auch weitere Bestandteile beinhalten, was jedoch nicht der Schwerpunkt dieses Artikels ist.

Die Entwicklung der *Produktlinienplattform* geschieht nicht ad hoc, sondern auf Basis eines *fachlichen Domänenmodells*, das die gemeinsamen Funktionalitäten beschreibt und das auf Basis der Anteile des *fachlichen Prozessmodells* bzw. der Anteile der *fachlichen Anforderungen* entwickelt wird, die für alle Produkte identisch sind.

5 Fazit und Ausblick

In Abbildung 4 sind alle Konzepte, die in diesem Artikel betrachtet wurden, in einem Modell zusammengefasst, d. h. SOA, BPM, komponentenbasierte und modellbasierte Softwareentwicklung sowie Software-Produktlinien.

Selbstverständlich besteht noch ein großes Erweiterungspotenzial dieses Modells sowohl in der inhaltlichen Tiefe als auch in der Breite. In der Tiefe könnten sicherlich bei jedem der Einzelaspekte zusätzliche Entitäten und Beziehungen ergänzt werden. In der Breite könnten sicherlich noch weitere Konzepte aus interessanten aktuellen Themenfeldern, wie z. B. Software-as-a-Service oder Grid Computing, integriert werden.

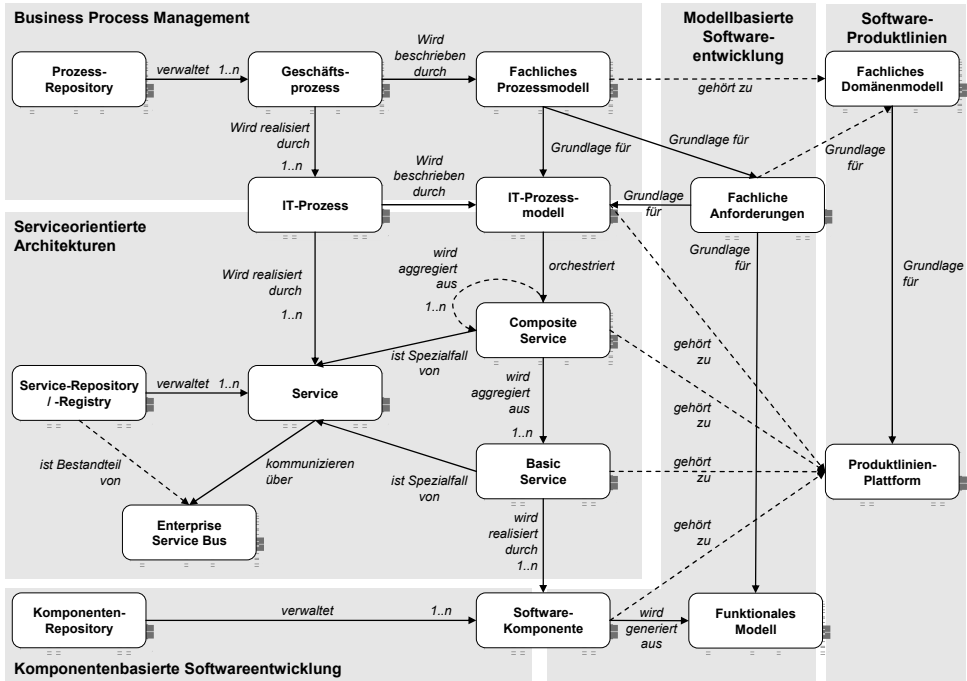


Abbildung 4: Integration von Software-Produktlinien

Zudem wurden in dem Beitrag organisatorische Aspekte der Industrialisierung der Softwareentwicklung weitestgehend ausgeklammert. Neben den hier beschriebenen primär technischen Aspekten dürfen diese in der Praxis jedoch nicht vernachlässigt werden.

Auch bei der Darstellung der einzelnen Konzepte bestehen noch Diskussionspotenziale. Der Beitrag sollte daher auch als Anregung für eine wissenschaftliche Diskussion dienen, die das Ziel besitzt, ein Begriffsnetz für aktuelle Themen im Bereich Software-Management und -Technik zu definieren.

Eine akzeptierte Ontologie in dieser Domäne würde sowohl der Wissenschaft als auch der Praxis helfen, oftmals bei Projekten auftretende langwierige und nicht zielführende Diskussion über einzelne Begriffe zu vermeiden.

Literaturverzeichnis

- [Am07] Amadee AG; Wolfgang Martin Team: Studie SOA Check 2007 – Trends im deutschen Markt. <http://www.soa-check.net>, zuletzt besucht am 23.5.2008.
- [Bi07] Biskup, H. et al.: Ein Begriffsnetz für die Software-Entwicklung. In: Informatik-Spektrum Band 30 Heft 3. Springer Verlag, Heidelberg, 2007; S. 217-224.
- [En08] Engels, G. et al.: Quasar Enterprise - Anwendungslandschaften serviceorientiert gestalten. dpunkt Verlag, Heidelberg, 2008.

- [Hö07a] Höß, O. et al.: Serviceorientierte Architekturen – Potenziale, Herausforderungen und Lösungsansätze. In (Spath, D. et al., Hrsg.): Serviceorientierte Architekturen. Fraunhofer IRB Verlag, Stuttgart, 2007; S. 1-11.
- [Hö07b] Höß, O. et al.: Migration zu serviceorientierten Architekturen - top-down oder bottom-up? In (Hildebrand, K., Hrsg.): IT-Integration & Migration. HMD - Praxis der Wirtschaftsinformatik, Heft 257. dpunkt Verlag, Heidelberg, 2007; S. 39-46.
- [Hö05] Höß, O.: Ein System für das Wiederverwendungs-Management von Software-Komponenten. Dissertation Universität Stuttgart. Jost-Jetter Verlag, Heimsheim, 2005.
- [Jo08] Josuttis, N.: SOA in der Praxis - System-Design für verteilte Geschäftsprozesse. dpunkt Verlag, Heidelberg, 2008.
- [LSR07] Linden, F. v. d.; Schmid, K.; Rommes, E.: Software Product Lines in Action - The Best Industrial Practice in Product Line Engineering. Springer Verlag, Heidelberg, 2007.
- [Oa06] OASIS: Reference Model for Service Oriented Architecture 1.0. <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>, zuletzt besucht am 23.5.2008.
- [PBL05] Pohl, K.; Böckle, G.; Linden, F. v. d.: Software Product Line Engineering: Foundations, Principles, and Techniques. Springer Verlag, Heidelberg, 2005.
- [Ro03] Rombach, D.: Software als Ingenieurprodukt. In (Warnecke, H.-J.; Bullinger, H.J., Hrsg.): Kunststück Innovation: Praxisbeispiele aus der Fraunhofer-Gesellschaft. Springer Verlag, Berlin, 2003; S. 75-82.
- [SI07] Stein, S; Ivanov, K.: EPK nach BPEL Transformation als Voraussetzung für praktische Umsetzung einer SOA. In (Bleek, W.-G.; Raasch, J.; Züllighoven, H., Hrsg.): Software Engineering 2007. Lecture Notes in Informatics (LNI), Band 105. Gesellschaft für Informatik (GI), 2007; S. 75-80.
- [Sp04] Spath, D.: Software als Innovationsmotor. In (Spath, D.; Haasis, K.; Klumpp, D., Hrsg.): Aktuelle Trends in der Softwareforschung. Tagungsband des doIT Software-Forschungstags 2004. Fraunhofer IRB Verlag, Stuttgart, 2004; S. 30-37.
- [SS07] Simon, M.; Schneider, C.: Strukturierung der EnBW SOA anhand eines Metamodells. In (Spath, D. et al., Hrsg.): Serviceorientierte Architekturen. Fraunhofer IRB Verlag, Stuttgart, 2007; S. 83-99.
- [SW08] Spath, D.; Weisbecker, A., Hrsg.: Business Process Management Tools 2008. Evaluierende Marktstudie. Fraunhofer IRB Verlag, Stuttgart, 2008.
- [SWH05] Spath, D.; Weisbecker, A.; Höß, O.: Modellbasierte Softwareentwicklung. In (Spath, D.; Weisbecker, A.; Höß, O., Hrsg.): Modellbasierte Softwareentwicklung. Fraunhofer IRB Verlag, Stuttgart, 2005; S. 11-14.
- [We02] Weisbecker, A.: Software-Management für komponentenbasierte Software-Entwicklung. Habilitation Universität Stuttgart. Jost-Jetter-Verlag, Heimsheim, 2002.