

Praxiserprobte Muster und Landkarten zur serviceorientierten Integration von Standardsoftware

Dr. Karl Prott, Markus Wissing

SOA Innovation Lab e.V.
c/o Deutsche Post AG
Charles-de-Gaulle-Straße 20
53113 Bonn

karl.prott@capgemini-sdm.com
markus_wissing@mckinsey.com

Abstract: Eine Arbeitsgruppe des SOA Innovation Lab e.V. (www.soa-lab.de) analysiert derzeit, welche Erfahrungen die verschiedenen Mitgliedsunternehmen mit der serviceorientierten Integration von Standardsoftware gemacht haben. Die Ergebnisse werden auf der Basis eines Reifegradmodells für serviceorientierte Architekturen ausgewertet und für alle Mitglieder aufbereitet. Als Teilaspekt wurde insbesondere die Frage untersucht, inwiefern sich die serviceorientierte Integration von Standardsoftware von den in vielen Unternehmen etablierten Integrationslösungen unterscheidet und welche Lösungsmuster sich in der Praxis als hilfreich erwiesen haben.

1 Einleitung und Motivation

Standardsoftwareprodukte koexistieren mit individuell entwickelten Anwendungen in der IT-Anwendungslandschaft und sind mit diesen auf vielfältige Weise verzahnt. Da beide Arten von Software jedoch unterschiedlichen Treibern folgen, entsteht ein Spannungsfeld: Individualsoftware wird in der Regel über Modularisierung und Serviceorientierung hinsichtlich Agilität, Erweiterbarkeit und Integrierbarkeit optimiert. Standardsoftwareprodukte im Enterprise-Umfeld hingegen sind hauptsächlich auf niedrige Total Cost of Ownership (TCO) und hohen Durchsatz ausgerichtet; Modularisierung und Serviceorientierung waren in der Vergangenheit nicht die entscheidenden Erfolgsfaktoren.

Für so genannte End-to-End-Geschäftsprozesse ist die Verzahnung der beteiligten Individual- und Standardsoftwareprodukte jedoch inzwischen ein entscheidender Erfolgsfaktor geworden. Von einer serviceorientierten Integration verspricht man sich das notwendige Maß an Agilität und Flexibilität, um sich am Markt von seinen Mitbewerbern differenzieren zu können ([En+08], [Kr+05], [Ke07]). Die Arbeitsgruppe entwickelt einen Werkzeugkasten, der Mitgliedsunternehmen dabei helfen soll, maßgeschneiderte Lösungen für diese Problemstellung zu finden. Zum Werkzeugkasten gehören eine Liste von bewährten Integrationsmustern (Kapitel 2) und eine Landkarte

der Integrationskomponenten (Kapitel 3). Komplexe Integrationslösungen (z.B. für End-to-End-Prozesse) kombinieren viele Integrationsmuster und -komponenten. Ziel der Arbeitsgruppe ist es, solche komplexen Lösungsmuster zu erarbeiten und mit Herstellern von Standardsoftware und Integrationslösungen zu erproben. Dabei wird die Arbeitsgruppe die von den Herstellern angebotenen Integrationslösungen auf der Landkarte der Integrationskomponenten abbilden und hinsichtlich herstellerübergreifender Muster analysieren (Kapitel 4).

2 Integrationsmuster

Seit Einführung der Implementierungsmuster durch die Gang of Four [Ga94] wurden Patterns für alle Ebenen der Softwarearchitektur entwickelt. Auch für die Gestaltung von ganzen Anwendungslandschaften liegen in der Literatur einige Beispiele vor, etwa [FP03], [Er09], [HW03]. Im Fokus der Arbeitsgruppe stand daher nicht das Finden neuer Enterprise Patterns, sondern das Identifizieren von Mustern oder Verfahrensweisen, die insbesondere für die serviceorientierte Integration von Standard- und Individualsoftware relevant sind und sich in der Unternehmenspraxis bewährt haben.

Damit alle an der Arbeitsgruppe beteiligten Unternehmen ihre Integrationsmuster einheitlich aufbereiten und so ihre Erfahrung miteinander teilen können, wurde zuerst ein einheitliches Dokumentationsformat entwickelt. Die in fünf Teile strukturierte Präsentation vorlage ermöglicht eine übersichtliche Darstellung von Integrationsmustern. Teil 1 ist ein Einseiter, der in sieben Punkten das Muster zusammenfassend skizziert: das zu lösende Problem, die Lösung selbst, ein Anwendungsbeispiel, verwandte Muster, Verwendungsempfehlung, Gegenanzeige für die Verwendung sowie Vor- und Nachteile der Lösung. Die Struktur orientiert sich an gängigen Musterbeschreibungen in der Literatur [Er09], [FP03] und [Ga94]. Die Teile 2 und 3 der Vorlage dienen dazu, das Problem auf jeweils mindestens einer Folie detailliert zu beschreiben und anhand eines konkreten Beispiels zu erläutern. In den Teilen 4 und 5 wird dann anhand des Beispiels das in der Praxis eingesetzte Integrationsmuster vorgestellt.

Bei der Erarbeitung eines initialen Katalogs von Integrationsmustern wurden drei Arten von Integrationsmustern erkannt: erstens etablierte Integrationsmuster wie die Service-Fassade, die eingesetzt werden, um die Anwendungslandschaft servicefähig zu machen. Als Zweites benötigt man Lösungsmuster für den Einsatz von typischen Integrationskomponenten wie ein Service Repository. Als Drittes resultierte aus der Analyse von typischen Anwendungsfällen die Erkenntnis, dass man bei der serviceorientierten Integration von Standardprodukten auf Probleme stößt, zu deren Lösung komplexe Integrationsmuster notwendig sind, die gegebenenfalls den kombinierten Einsatz mehrerer einfacher Muster erfordern.

2.1 Service-Fassade

Unternehmen nutzen verschiedene Patterns, um die Anwendungslandschaft servicefähig zu machen. Ein geeignetes Pattern ist die Service-Fassade zur Kapselung von heterogenen Anwendungen.

In einem IT-Projekt zur Optimierung von Finanzierungsprozessen wurde u.a. der Teilprozess für das Zinsfixing von so genannten Rollover-Krediten überarbeitet. Vor der Neuausrichtung mussten mehrere Prozessbeteiligte in unterschiedlichen Rollen bei jedem Prozessschritt verschiedene Anwendungen mit sehr unterschiedlichem "Look and Feel" bedienen. Doppel­eingaben waren notwendig. Drohende Fehler und Terminüberschreitungen ließen sich nur durch eine aufwändige Koordination aller Prozessbeteiligten vermeiden.

Die serviceorientierte Integration mittels einer Service-Fassade hat zu deutlichen Verbesserungen geführt – es sind wiederverwendbare, konfigurierbare und orchestrierbare Service-Operationen mit einheitlichem "Look and Feel" über alle Prozessschritte entstanden. Die Service-Fassade kapselt technisch und fachlich sehr unterschiedliche Individual- und Standardsoftwarekomponenten (s. Abbildung 1). In Kombination mit einer Aufgabenliste und einer Prozesssteuerung wird der Endnutzer von automatisierbaren Prozessschritten wie beispielsweise der Benachrichtigung der nächsten am Prozess beteiligten Person(en) entbunden. Insgesamt hat sich die Prozessqualität spürbar erhöht, Doppel­eingaben können entfallen, die Prozessüberwachung erfolgt automatisch. Nicht zuletzt lassen sich Mitarbeiter durch die Prozessharmonisierung nun flexibler einsetzen als zuvor.

So oder ähnlich stellt sich der Einsatz der Service-Fassade zur Integration von Standard- und Individualsoftware in einem durchgängigen Prozess in vielen Unternehmen dar. Damit kann die Service-Fassade als ein in der Praxis bewährtes Integrationsmuster gesehen werden.

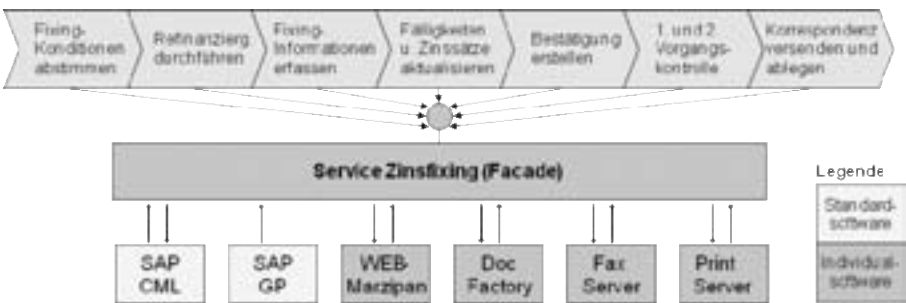


Abbildung 1: Integrationsmuster Service-Fassade

2.2 Service Repository

Eine sinnvolle Komponente für den serviceorientierten Aufbau der Anwendungslandschaft (vgl. Abbildung 2) ist ein Service Repository. Dabei trennt man gerne in ein Repository für die Projektphasen bis zur Entwicklung und eines für die Laufzeit (inklusive Test). Letzteres ist dann häufig eine Registry (z.B. UDDI). Da viele Standardsoftwareprodukte ihr eigenes Service Repository mitbringen, haben Unternehmen oft gleichzeitig verschiedene Service Repositories im Einsatz. Meist sind diese jedoch nicht kompatibel.

Für Unternehmen stellen sich damit in der Regel entscheidende Fragen wie

- Was ist der beste Weg, alle Service Repositories in einer logischen Sicht zu integrieren? Rechnet sich der Aufwand überhaupt?
- Lassen sich die unterschiedlichen Bedürfnisse z.B. in puncto Entwicklungs- und Laufzeit mit einem Repository erfüllen? Falls ja, ist das überhaupt sinnvoll?

Bei vielen Unternehmen sind diese Fragen noch ungeklärt, eine zentrale Sicht auf alle verfügbaren Services fehlt. Aufbauend auf den Erfahrungen der Mitgliedsunternehmen möchte die Arbeitsgruppe des SOA Innovation Lab ein Integrationsmuster entwickeln, mit dem sich diese Fragen beantworten lassen.

2.3 Komplexe Integrationsmuster

Bei der serviceorientierten Integration von Standardprodukten ergeben sich häufig Probleme, die sich nur mit komplexeren Integrationsmustern lösen lassen, die gegebenenfalls den kombinierten Einsatz mehrerer einfacher Muster erfordern. So stellte ein Unternehmen bei der Analyse seines Auftragsabwicklungsprozesses (Order-to-Cash) fest, dass z.B. Kundeninformationen in vielen Prozessschritten und damit in unterschiedlichsten Anwendungen benötigt werden. Die einzelnen Anwendungen bieten dabei eine unterschiedliche Sicht auf den Kunden, was eine Synchronisation der Daten erschwert. Die Kundendaten müssen in unterschiedlichen Systemen gepflegt werden. Erschwerend kommt hinzu, dass manche Produkte (z.B. SAP GP/CRM) beanspruchen, Master zu sein, obwohl das Unternehmen die Stammdaten in einer anderen Anwendung pflegen möchte. Eine Synchronisation der Daten – bei verschiedenen Produkten eines Herstellers oft noch möglich – ist unter diesen Umständen extrem aufwändig oder kann sogar scheitern, wenn die beteiligten Anwendungen zu heterogen sind. Die Folge: Eine zentrale Stammdatensicht ist nicht verfügbar. Auch der Einsatz eines zentralen technischen Stammdatenprodukts, wie es viele MDM-Produkthersteller anpreisen, bietet hier keine einfache Lösung.

Erste Diskussionen der Arbeitsgruppe zeigen, dass Unternehmen unterschiedliche Integrationsmuster auf dem Weg zu einer zentralen Stammdatensicht implementieren. So wurde über eine Service-Fassadenlösung diskutiert, in der z.B. die Implementierung eines Service "Kunde" in der Fassade die Verwaltung der Kundendaten in allen beteiligten Systemen koordiniert. Andere Lösungen beinhalteten die Einführung von intelligenten Caching- und Synchronisationslösungen. So setzt sich eine Lösungsempfehlung aus der Kombination von verschiedenen einfacheren Integrationsmustern zusammen.

3 Landkarte der Integrationskomponenten

Die Landkarte der Integrationskomponenten wurde von den Mitgliedern der Arbeitsgruppe als Referenzarchitektur für die Integration von Geschäftsanwendungen entwickelt. Sie enthält und gruppiert alle technischen Integrationsfunktionen, die man u.a. zur Umsetzung einer SOA benötigt.

Integrationskomponenten werden als etablierte Bausteine einer Integrationslandschaft verstanden. Integrationsmuster liegen orthogonal dazu und werden zwecks Integration von Geschäftsanwendungen mit Integrationskomponenten zu einer Gesamtintegrationslösung kombiniert. Zur Strukturierung dieser Bausteine haben sich die Integrationsebenen Präsentation (Frontend), Prozess, Logik und Daten [En+08] herauskristallisiert. Neben Komponenten, die über Integrationsebenen hinweg relevant sind, wurden Komponenten für Entwicklung und Laufzeit identifiziert.

Die Anwendung der Landkarte unterstützt eine ganzheitliche Betrachtung der Integrationsproblematik. Neben der Logikintegration, die meist im Vordergrund einer SOA steht, ist die Datenintegration eine weitere Voraussetzung (z.B. Master Data Management) für jegliche Logik- und Prozessintegration. Prozess- und Präsentationsintegration liefern höherwertige Integrationslösungen. Des Weiteren lassen sich mit der Landkarte Herstellerlösungen im Bereich Integration und Middleware sehr gut analysieren, vergleichen und bewerten [He+08]. Ein willkommener Nebeneffekt ist die Möglichkeit, mit der Landkarte Überlappungen im Lösungsportfolio des Herstellers zu erkennen.

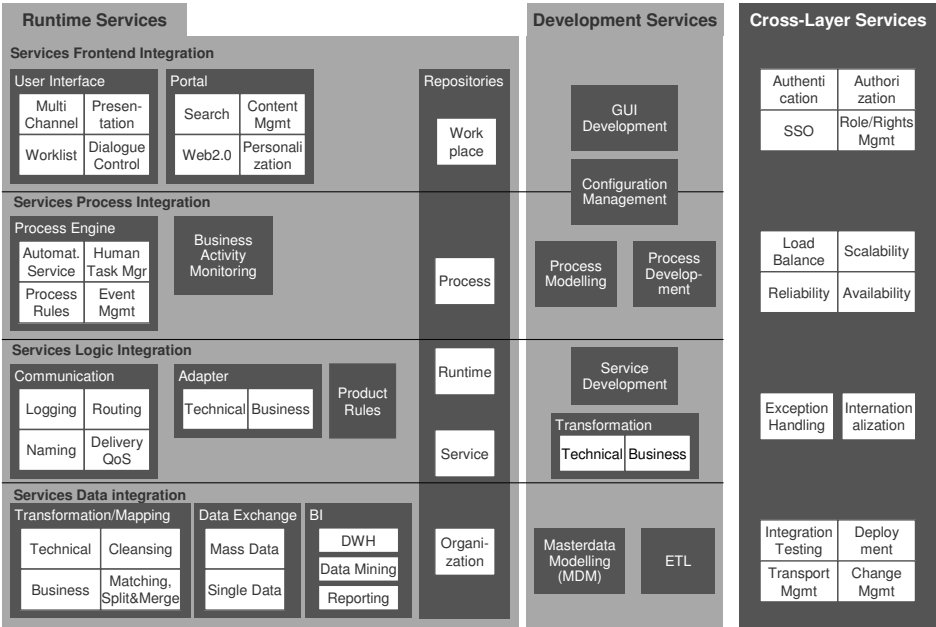


Abbildung 2: Landkarte der Integrationskomponenten

4 Fazit und Ausblick

Derzeit wenden die Mitglieder der Arbeitsgruppe die beschriebenen Integrationsmuster und die Landkarte der Integrationskomponenten auf konkrete Probleme rund um die Einbindung von Standardsoftware in End-to-End-Prozessen an. Master Data Management und Business Activity Monitoring scheinen dabei zwei hochpriorisierte Problemstellungen zu sein. Aufbauend auf diesen Anwendungsfällen sollen konkrete Fragen und Anforderungen an die Hersteller von Standardsoftware und Integrationslösungen abgeleitet und Lösungsempfehlungen in einer Testumgebung erprobt werden. Als Ergebnis sind komplexe Integrationsmuster zu beschreiben.

In einem weiteren, in diesem Artikel nicht näher betrachteten Schritt hat die Arbeitsgruppe den Reifegrad der serviceorientierten Integrierbarkeit von Standardsoftwareprodukten verschiedener Hersteller untersucht und bewertet. Hierzu wurde ein Reifegradmodell entwickelt, das sich an bekannten Reifegradmodellen ([Ma09]) und Architektur-Frameworks ([To09], [IAF]) orientiert.

Literaturverzeichnis

- [En+08] Engels, G.; Hess, A., Humm, B.; Juwig, O.; Lohmann, M.; Richter, J.P.; Voß, M.; Willkomm, J.: Quasar Enterprise. dpunkt.verlag 2008.
- [Er09] Erl, T.: SOA Design Patterns. Prentice Hall 2009.
- [FP03] Fowler, M.: Patterns of Enterprise Application Architecture. Addison Wesley 2003.
- [Ga94] Gamma E.; Helm R.; Johnson R.; Vlissides J.: Design Patterns. Elements of Reusable Object-Oriented Software. Addison Wesley 1994, ISBN 0-201-63361-2.
- [He+08] Heinisch, M.; Kölliker, M.; Könings, M.; Lattmann, M.; Liebhart, D.; Pakull, P.; Schmutz, G.; Welkenbach, P.: Integration Architecture Blueprint. Hanser Verlag 2008.
- [HW03] Hohpe, G.; Woolf, B.: Enterprise Integration Patterns. Addison Wesley 2003.
- [IAF] www.capgemini.com/iaf.
- [Ke07] Keller, W.: IT-Unternehmensarchitektur. dpunkt.Verlag 2007.
- [Kr+05] Krafzig, D.; Banke, K.; Slama, D.: Enterprise SOA. Prentice Hall 2005.
- [Ma09] State of the Art SOA Maturity Models:
- ACMM Architecture Capability Maturity Model. The Open Group 2009.
 - Inaganti, S., Aravamudan, S.: SOA Maturity Model. BP Trends, April 2007. <http://www.bptrends.com/publicationfiles/04-07-ART-The%20SOA%20MaturityModel-Inagantifinal.pdf>, 2009.
 - Sonic: SOA Maturity Model. http://soa.omg.org/Uploaded%20Docs/SOA/SOA_Maturity.pdf, 2009.
 - Oracle: SOA Maturity Model. <http://www.scribd.com/doc/2890015/oraclesoamaturitymodelcheatsheet>, 2009.
 - Open Group: OSIMM Maturity Model for SOA. <http://www.opengroup.org/projects/soa-book/page.tpl?CALLER=faq.tpl&ggid=1319>, 2009.
 - IBM: SIMM Services Integration Maturity Model. <http://www.ibm.com/developerworks/webservices/library/ws-soa-simm/>, 2009.
- [To09] TOGAF Version 9. The Open Group Architecture Framework 2009