

BPMN trifft auf SOA-Governance

Arne Brüning, ORACLE Deutschland B.V. & Co. KG

Sobald ein Unternehmen SOA-Technologien auf mehrere Projekte ausrollt, sollte auch über einen ersten Ansatz zur SOA-Governance nachgedacht werden. Durch die Leistungsfähigkeit der Tools zur Geschäftsprozessmodellierung lassen sich diese Prozesse vermehrt direkt auf IT-Systemen ausführen. Um auch Sinnvolles zu verrichten, benötigen die Prozesse oftmals Zugriff auf die Services der SOA. Dieser Artikel zeigt einen Ansatz, wie die Abstimmung zwischen den fachlich motivierten Prozess-Entwicklern und SOA-Architekten über das Oracle Enterprise Repository erfolgen kann.

Viele Unternehmen haben heute den opportunistischen Ansatz zum Einstieg in die SOA gewählt und überschaubare Projekte in kleinen Teams auf Zuruf realisiert. Kommt jetzt aber die Erfolgskrise und soll die erfolgreiche neue SOA-Technologie auf weitere Projekte oder Systeme ausgerollt werden, stellt sich schnell heraus, dass mehr Ordnung im Prozess der Erstellung erforderlich ist, weil das System „auf Zuruf“ nicht mehr funktioniert.

Das Schlagwort lautet „SOA-Governance“. Bereits nach kurzer Recherche wird aber klar, dass daraus schnell eine Herkules-Aufgabe werden kann. Es gibt SOA-Governance-Modelle, die mehr Beteiligte beschreiben, als ein mittelständisches Unternehmen Mitarbeiter in der Entwicklung hat. Wünschenswert wäre hier oft eine etwas kleinere Lösung.

Ein wichtiger Aspekt, den die SOA-Governance regeln soll, ist die Wiederverwendung von Services. Für viele war ja ein Grund, sich mit SOA-Technologien zu beschäftigen, das Nutzenversprechen der Wiederverwendung. Aber wie bei vielen früheren Technologien, die mit der gleichen Zusage antraten, stellt sich auch bei SOA die Wiederverwendung nicht automatisch ein. Eines der Haupt-Hindernisse war schon immer das „not-invented-here“-Syndrom. Manchen Entwicklern fällt es schwer, Komponenten aus anderen Abteilungen zu vertrauen, wenn sich nicht sicherstellen lässt, dass diese den eigenen Anforderungen bezüglich Qualität, Sicherheit, Performance etc. genügen. Auch steht der Ein-

sparung durch den wiederverwendeten Code ein Mehraufwand an Abstimmung gegenüber, der sich für den Entwickler gegebenenfalls schlechter abschätzen lässt, als der Aufwand, die entsprechende Komponente gemäß den eigenen Anforderungen neu zu entwickeln. Aus Sicht der SOA-Architekten wird hier ein Verfahren benötigt, das Standards und Abläufe definiert, um die Hemmnisse der Wiederverwendung abzubauen.

Über die Service-Inventarisierung in einem zentralen Repository wie dem Oracle Enterprise Repository lässt sich mit einem überschaubaren Aufwand bereits einiges an Nutzen generieren. Sobald man mit der Inventarisierung beginnt, stellen sich erste Fragen, die geklärt werden müssen. Wer darf etwas in das Repository einstellen? Wie müssen die Services mindestens beschrieben werden? Welche qualitativen Anforderungen müssen Services erfüllen, um ins Repository aufgenommen zu werden? Wie sehen die Freigabeverfahren aus? Sobald diese Fragen geklärt sind, hat die IT-Abteilung den Einstieg in die persönliche, maßgeschneiderte SOA-Governance getan. Die damit definierten Standards und die damit verbundenen Prozesse zu deren Einhaltung wirken sich begünstigend auf die Wiederverwendung aus.

Die Sicht der Prozess-Entwickler

Die Prozess-Entwickler interessieren sich für diese IT-Belange zunächst wenig. Wenn sie aber Prozesse erstellen möchten, die anschließend von der IT ausgeführt werden

sollen, benötigen sie Zugriff auf die entsprechenden Services. Diese sollen später zur Laufzeit die von den Prozessen oft nur abstrakt definierten Aufgaben verrichten. Hierzu sind Mitarbeiter erforderlich, die sowohl die Prozessmodellierung als auch die Services so weit verstehen, dass sie diese den Aktivitäten im Prozess zuordnen können.

Natürlich lässt sich der Prozess zunächst unter rein fachlichen Gesichtspunkten modellieren und die IT kann sich später um die Realisierung kümmern. Werden aber die Anforderungen aus der Prozess-Sicht überhaupt nicht gegen das Service-Portfolio abgeglichen, besteht die Gefahr, dass das Projekt in einer späten Phase ins Stocken gerät, wenn eben diese Anforderungen überhaupt nicht in die Portfolio-Planung der SOA-Architekten passen. Im Sinne einer reibungslosen Zusammenarbeit sollte daher bereits möglichst frühzeitig eine Abstimmung erfolgen.

Aus Sicht des Prozess-Designers sieht die ideale Lösung vielleicht so aus: Er hinterlegt in einem System eine in Freitext formulierte Anfrage wie: „Für einen Prozess zur Genehmigung von Urlaubstagen benötige ich einen Dienst, der mir zu einer gegebenen Personalnummer die Anzahl der verbleibenden Resturlaubstage liefert“. Als Ergebnis bekommt er dann eine Aussage, ob so ein Dienst bereits existiert oder erstellt werden kann. Er kann auch ein technisches Artefakt für seinen Prozess bekommen, welches er bei Bedarf direkt in seinen Prozess integrieren kann.

Das Oracle Enterprise Repository

Nahezu beliebige Artefakte zu speichern und miteinander in Beziehung zu setzen ist eine der Hauptaufgaben des Oracle Enterprise Repository (OER). Dazu verfügt es über ein flexibles Datenmodell, das der Anwender an die eigenen Bedürfnisse anpassen kann. Typischerweise erfolgt die Datenhaltung in einer Oracle-Datenbank, die Ausführung der eigentlichen Anwendung auf einem WebLogic-Server. Es werden aber auch andere relationale Datenbanken und JEE-Application-Server unterstützt.

Warum sollte man nun aber überhaupt ein Repository einsetzen? Manche IT-Abteilungen setzen beispielsweise ein Wiki auf und pflegen die vorhandenen Services dort ein. Natürlich lassen sich grundsätzlich Services auch auf diese Art inventarisieren, es wird einfach pro Service ein beschreibender Text vergeben und für die Übersicht oder Suche sorgt das jeweilige Wiki-Framework. Der Nachteil einer solchen Lösung ist, dass es sich hierbei um unstrukturierte Informationen handelt und keinerlei weiter-

gehende Information zur Service-Inventarisierung enthalten ist. Wenn beispielsweise ein fertig entwickelter Service „live gehen“ soll, möchte man ihn möglicherweise automatisch in der UDDI-Registry oder beim Service Bus anmelden. Das ließe sich zwar über entsprechende Templates im Wiki und einem geeigneten, selbst zu entwickelnden Scripting zur Not noch implementieren. Aber die Liste der möglicherweise benötigten Funktionalitäten ist ja noch länger. Wenn die Freigabeverfahren komplexer werden, etwa weil außer den SOA-Architekturen bei bestimmten Services auch noch der Datenschutzbeauftragte eingebunden werden muss und für unterschiedliche Projekte auch unterschiedliche Gruppen von Genehmigern erforderlich sind, wird eine Workflow- oder Prozess-Komponente benötigt. Nun ließe sich sicherlich auch eine Workflow-Engine an ein Wiki anbinden, aber an diesem Beispiel wird klar, dass daraus schnell ein eigenes Entwicklungsprojekt wird. Das Oracle Enterprise Repository bringt hingegen eine entsprechende Funk-

tionalität bereits mit. Eine Synchronisation mit der Oracle Service Registry und auf diesem Weg auch mit dem Oracle Service Bus gehört zum Lieferumfang. Einmal aufgesetzt arbeitet diese transparent im Hintergrund. Auch mehrstufige Freigabeverfahren sind möglich, für komplexere Workflows liegt die hauseigene BPM-Engine bei.

Eine andere Anforderung aus der Praxis besteht darin, dass die Services nur genutzt werden dürfen, wenn der Anwender seine betreffenden Server vorher im Oracle Enterprise Repository als Konsumenten registriert hat, um die Benutzung nachvollziehen zu können. Der Lösungsansatz hierfür war, im Oracle Enterprise Repository die Verbindung zu den konsumierenden Servern vorzusehen. Im Betrieb werden die Services dann netzwerkseitig so abgeschottet, dass sie nur über einen Proxy mit entsprechender Filterfunktion zugreifbar sind, etwa über das Oracle Enterprise Gateway. Registriert sich nun ein Anwender für einen Service im Repository, so kann er die Server, von denen aus er den Service ver-



Schneller zum Wesentlichen!

Einfach, verständlich, vollständig: Mit HUNKLER machen Sie Business Intelligence vom Start weg richtig.

- Integrierte, optimal abgestimmte Komplettlösungen für jeden Bedarf
- Zielgruppengenaue Reportvorlagen
- Robuste Technologiebasis (z. B. Oracle BI Server, Oracle Data Integrator)
- Stark verkürzte Projektzeiten
- Flexibel, skalierbar, investitionssicher
- Spezielle Lösung für SAP R/3
- Kooperation mit SAP-Spezialist NewFrontiers (www.newfrontiers.com)

ORACLE Platinum Partner

Partner von
NewFrontiers
10 Years!

Best Solutions Based on Oracle
HUNKLER
GmbH & Co. KG

Hauptsitz Karlsruhe
Geschäftsstelle Bodensee

Bannwaldallee 32
Fritz-Reichle-Ring 2

76185 Karlsruhe
78315 Radolfzell

Tel. 0721-490 16-0
Tel. 07732-939 14-00

Fax 0721-490 16-29
Fax 07732-939 14-04

info@hunkler.de
www.hunkler.de

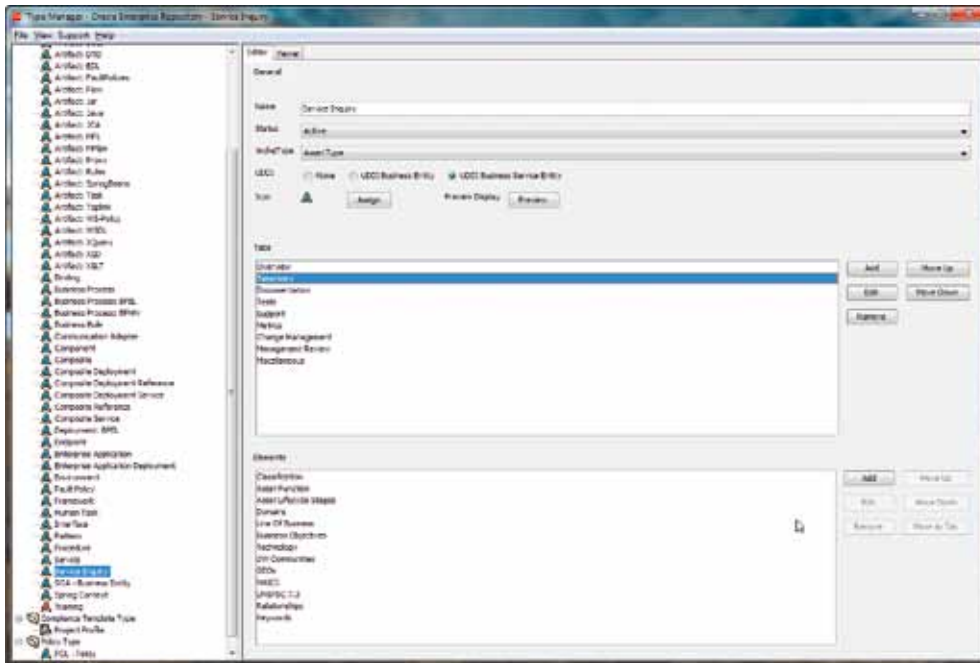


Abbildung 1: Die Service Inquiry im Type Manager

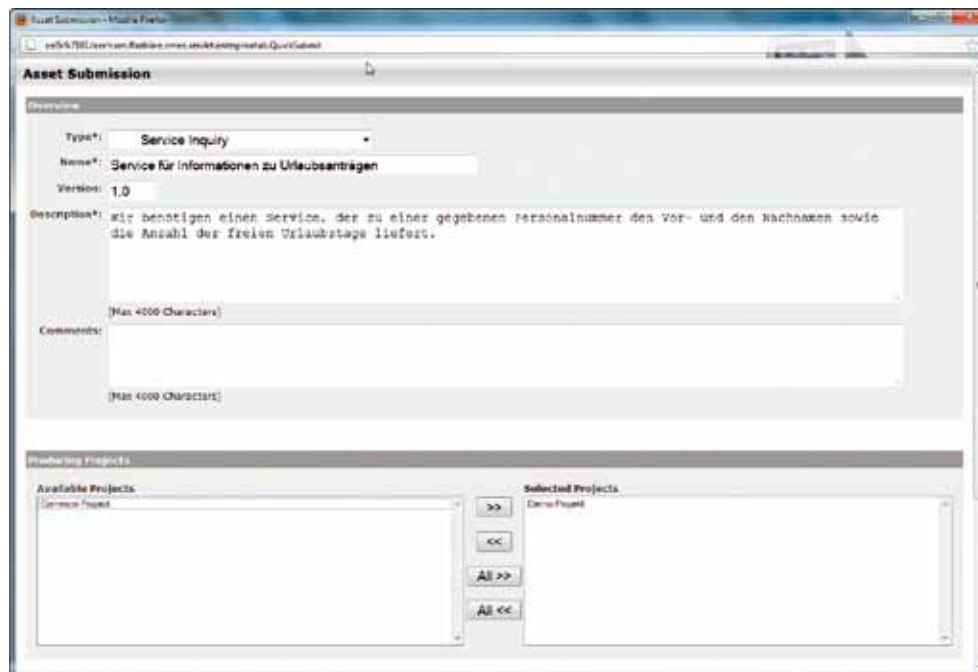


Abbildung 2: Die ausgefüllte Service Inquiry

wenden will, im Repository hinterlegen. Im Hintergrund sammelt ein Prozess nun die hinterlegten Server und aktualisiert die Whitelist des Proxys entsprechend. Auf diesem Weg wird vermieden, dass jemand die Services am Repository vorbei nutzt.

Auch zur Unterstützung der Wiederverwendung bietet das Oracle Enterprise Repository spezielle Funktionen an. So gibt es beispielsweise eine Bewertungsfunk-

tion ähnlich wie bei bekannten Webshops. Erfahrungen zeigen dort, dass sich Artikel mit guten Kundenbewertungen auch besser verkaufen. Diesen Effekt macht sich das Oracle Enterprise Repository zunutze und gibt den Anwendern die Gelegenheit, über eine Bewertung von einem bis zu fünf Sternen sowie einem Freitext für Kommentare die angebotenen Services zu beurteilen.

Zusätzlich zu den Benutzerkommentaren lassen sich auch noch technische Aspekte zur Beurteilung der Qualität von angebotenen Diensten hinzuziehen. So integriert sich das Oracle Enterprise Repository auch mit dem Enterprise Manager Grid Control und ermöglicht auf diesem Wege die Übernahme von Laufzeit-Informationen. So lassen sich, entsprechende Konfiguration vorausgesetzt, zu einem Service beispielsweise die durchschnittliche Verfügbarkeit, die Antwortzeit oder der Erfüllungsgrad der Service-Level-Agreements darstellen.

Um jetzt auf das eingangs erwähnte Beispiel des Entwicklers zurückzukommen, der vor der Entscheidung steht, einen Dienst wiederzuverwenden oder selbst zu implementieren, dann kann das Oracle Enterprise Repository hier eine Unterstützung sein. Wenn der Entwickler in einem gut gepflegten Repository schnell einen geeigneten Dienst findet, dort sehen kann, dass die Einhaltung der technischen Standards überprüft wurde und das Laufzeitverhalten einwandfrei ist, dieser Service dann möglicherweise noch eine gute Bewertung von anderen Anwendern erhalten hat, dann wird er sich mit Sicherheit leichter für die Wiederverwendung entscheiden, als wenn er diese Informationen nicht hätte.

Das Zusammenspiel von SOA und BPMN

Für den Anwender bietet das Oracle Enterprise Repository eine zweigeteilte Oberfläche. Möchte man nur bestehende Services nutzen oder administrative Aufgaben durchführen, reicht das HTML-Frontend aus. Zum Erstellen oder Bearbeiten wird der Asset-Editor genutzt, der in Java implementiert ist und aus dem HTML-Frontend via Java-Web-Start geöffnet wird. Damit lassen sich nicht nur die eigentlichen Artefakte bearbeiten, sondern auch die verschiedenen Artefakt- oder Asset-Typen sowie ihre zulässigen Parameter und ihre Darstellung im Asset-Editor festlegen. Damit nun das SOA-Team nach Installation des OER nicht die nächsten Tage mit Setup-Aufgaben verbringen muss, liefert Oracle das Repository mit einer Vielzahl von vordefinierten Typen und Attributen aus, sodass bereits die Standard-Installation die Anforderungen des SOA-Teams sehr gut abbildet. Für die Kommunikation mit den Prozess-Entwick-

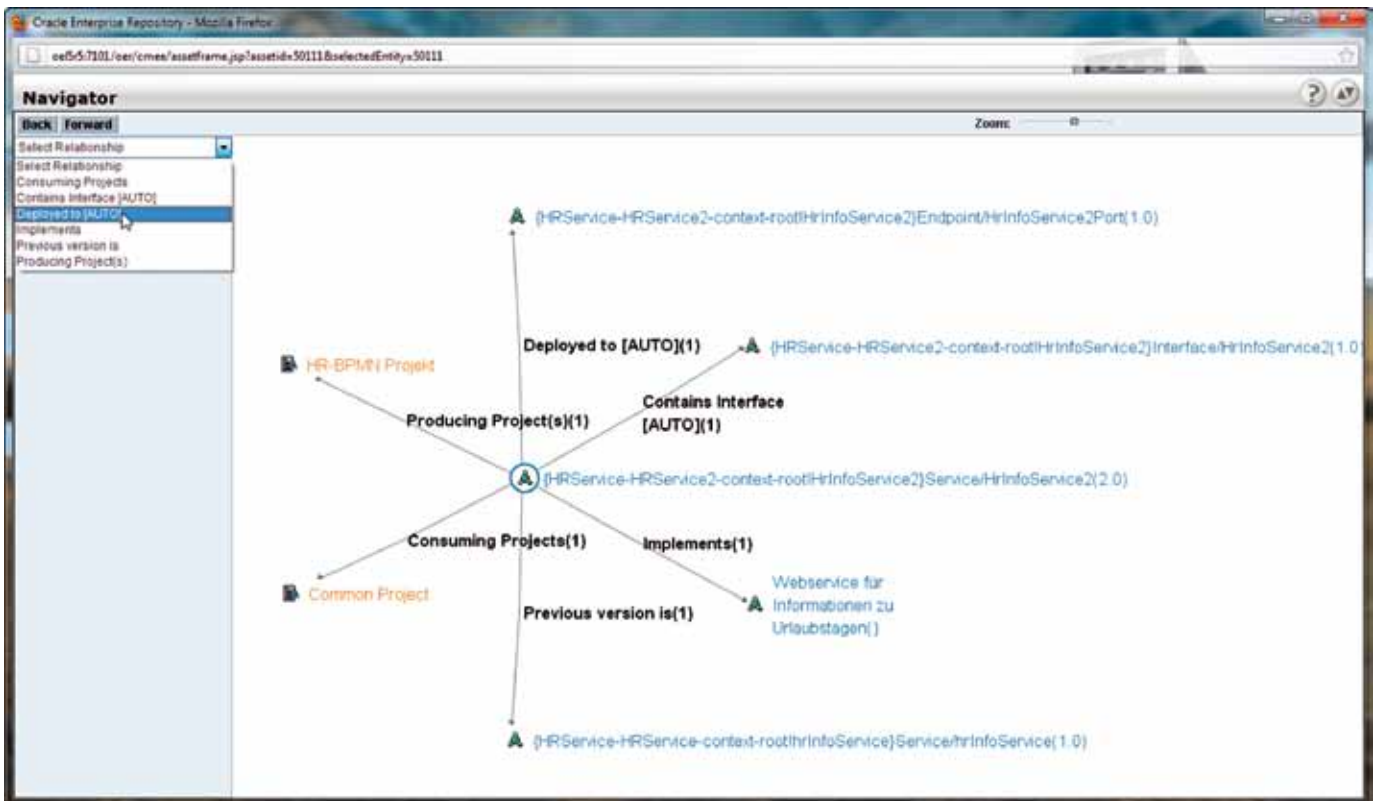


Abbildung 3: Die Beziehungen im Repository

lern ist ab Werk noch keine Funktionalität vorgesehen. Diese lässt sich aber sehr leicht einrichten, der Artikel zeigt im Folgenden einen möglichen Weg auf.

Die Flexibilität des Datenmodells und des Asset-Editors sollen genutzt werden, um das Zusammenspiel von SOA- und Prozess-Team zu unterstützen. Die Grundidee hierzu ist, dass ein Prozess-Entwickler zunächst eine einfache, umgangssprachliche Anfrage in das Repository einstellt. Hierzu lässt sich ein neuer Asset-Typ im Repository definieren, den man beispielsweise „Service Anfrage“ oder „Service Inquiry“ nennt.

Im OER können Asset-Typen von bestehenden Typen vererbt werden. In diesem Fall wird die „Service Inquiry“ vom normalen Service abgeleitet. Da aber ein Großteil der technischen Attribute für die „Service Inquiry“ nicht notwendig ist und dort eher stören würde, sollten diese entfernt werden. Für jeden Asset-Typ kann im OER festgelegt werden, wie dieser Typ später angezeigt werden soll. Hierzu können bestehende Tab-Reiter entfernt oder auch neue hinzugefügt werden (siehe Abbildung 1).

Der neu erzeugte Asset-Typ „Service Inquiry“ kann jetzt zum Erzeugen von Assets

genutzt werden wie jeder andere Typ auch. Der Prozess-Entwickler kann sich nun über die Standardoberfläche am Repository anmelden und seine Anfrage im Repository erfassen (siehe Abbildung 2).

Ein SOA-Architekt kann sich automatisch bei neuen Anfragen im Repository benachrichtigen lassen. Er kann jetzt prüfen, ob für die Anfrage ein neuer Service entwickelt werden muss oder ob sie idealerweise durch einen bestehenden Service gelöst werden kann, auch wenn dieser hierfür weiterentwickelt werden muss.

Sobald ein Service zur Verfügung steht, kann dies über eine entsprechende Relation im Repository dargestellt werden. Im Oracle Enterprise Repository können diese Abhängigkeiten auch grafisch in einer sogenannten „Spider-Web“-Darstellung visualisiert werden (siehe Abbildung 3). Der Prozess-Entwickler kann nun im JDeveloper über das Enterprise Repository Plug-in den ihm neu zur Verfügung gestellten Service direkt in seinen Prozess einbinden, während er die technischen Aspekte des Prozesses ergänzt.

Auch wenn dieses Beispiel bewusst simpel gehalten ist, zeigt es doch auf, dass ein

Repository nicht nur technische Artefakte enthalten muss. Gerade wenn eine Anfrage nach einem Service im Repository abgelegt wird, lässt sich der Verlauf nachvollziehen. Warum wurde ein bestehender Service neu beziehungsweise auf welche Anfrage hin wurde ein Service neu entwickelt? Dies ist nicht nur zu dokumentarischen Zwecken nützlich, sondern kann auch helfen, später einmal nachzuvollziehen, warum eine Änderung überhaupt vorgenommen wurde, wenn etwa einmal ein Re-Design ansteht.

Fazit

Für komplexere Szenarien liefert Oracle mit dem Enterprise Repository auch die haus-eigene BPMN-Prozess-Engine mit, sodass sich auch sehr komplexe SOA-Governance-Prozesse damit abbilden lassen. Wenn dann trotz aller Werkzeugunterstützung die Beteiligten auch noch miteinander reden, ist der erste Schritt in Richtung eines betriebsinternen, maßgeschneiderten SOA-Governance-Modells getan.

Arne Brüning

ORACLE Deutschland B.V. & Co. KG

arne.brueuning@oracle.com