

Modellgetriebene Software-Entwicklung mit BPMN und SOA

Evgenia Rosa
Oracle Deutschland
Christian Dedek
Orientation in Objects GmbH
Mannheim

Schlüsselworte:

BPM, MDSO, Oracle SOA-Suite 11g , Oracle BPA-Suite 11g, BPMN, Human Task , ARIS Design Platform

MDSO meets BPMN Vision

Die Kombination aus modellgetriebener Softwareentwicklung (MDSO) und Business Process Management (BPM) bietet die Möglichkeit die Qualitäts- und Wartbarkeitsvorteile modellgetriebener Softwareentwicklung auch in die Welt von BPM-Lösungen zu übertragen. An dieser Kombination werden insbesondere die hohen Investitionen in Analyse, Modellierung und architekturelle Infrastruktur als eher dokumentationszentriert und weniger produktzentriert kritisch diskutiert. Die Frage nach einer wirtschaftlich sinnvollen Durchführbarkeit des gesamten Vorgehens ist wegen dieser Fokussierung auf produktfremden Artefakte daher besonders mit Blick auf Umsetzungsrisiko und -kosten eines Entwicklungsprojektes zu stellen.

Explizite fachliche Prozessmodelle in BPMN bieten einen interessanten Ausgangspunkt als verwertbare technische Artefakte für eine SOA-basierte Softwaregenerierung. Diese Modelle allein bieten keine ausreichende Basis für eine MDSO. Sie müssen mit weiteren Modellen und Diagrammen sowie einer Laufzeitarchitekturplattform kombiniert werden, um einen nennenswert generierbaren Systemanteil zu erzielen.

Mit der Oracle 11g SOA Suite existiert eine derartige JEE-basierte Laufzeitarchitekturplattform. Bereits seit der Version 10 bietet Oracle die BPA Suite als einen Werkzeugsatz für MDSO aus BPMN Modellen mit der SOA Suite als Zielplattform an. Die aktuell vorhandenen Modellierungsangebote dieses Stacks bieten bereits Möglichkeiten der Erstellung von technisch verwertbaren Artefakten zur generativen Weiterverwendung im Zielprodukt. Leider reichen diese ursprünglichen Basismodellierungstechniken der Oracle BPA Suite für eine wirtschaftliche ausnutzbare Softwaregeneration nicht aus. Insbesondere die Themenfelder Userinterface und Daten verschließen sich einer leichten Generierbarkeit aus den existierenden Modelltypen. Die Erweiterbarkeit der BPA Suite und die enge Integration mit den Konzepten der Laufzeitarchitektur der Oracle SOA Suite 11g erlauben jedoch die erfolgreiche Erweiterung der Basismodelle mit dem Ziel einer Softwaregenerierung.

Der Vortrag beschreibt die in einem Entwicklungsprojekt gemachten Erfahrungen mit der Generierung von Datenobjekten, Services, Benutzeroberflächen und Prozessen aus der Oracle BPA Suite. Dabei werden sowohl Standardmodelle der BPA Suite als auch projektspezifische Erweiterungen betrachtet.

Kurzvorstellung Oracle BPA/SOA-Suite Integration

Die Oracle SOA Suite enthält alle Serviceinfrastrukturkomponenten, die für Aufbau, Bereitstellung und Management einer SOA erforderlich sind. Mit der Oracle SOA Suite lassen sich Services erstellen, verwalten und zu modularen Anwendungen und Geschäftsprozessen zusammenstellen.

Die Oracle Business Process Analysis (BPA) Suite bietet Werkzeuge und Modelle zur beschleunigten Prozessverbesserung durch Konvertierung von Prozessmodellen in ausführbare Architekturelemente innerhalb der Oracle SOA Suite Laufzeitplattform. Die Technische Basis bildet dabei die IDS Scheer ARIS Design Platform mit ihren Modellen und integrierten Produkten für Design, Simulation und Optimierung von Geschäftsprozessen. Ziel der Kombination von BPA und SOA Suite ist es die Lücke von der strategischen Konzeption bis zur Umsetzung von Geschäftsprozessen im Unternehmen zu schließen.

Gemäß der Model Driven Architecture (MDA) der Object Management Group (OMG) werden drei Abstraktionsschichten der Modellierung unterschieden:

- CIM – Computation Independent Modeling (Konzeptionelle Schicht)
- PIM – Platform Independent Modeling (Logische Schicht)
- PSM – Platform Specific Modeling (Physikalische Schicht)

Die Oracle BPA Suite bietet Diagramm- und Modelltypen für den Einsatz in allen drei Schichten. Dabei unterstützen Verknüpfungsmöglichkeiten zwischen Modellartefakten die durchgängige Modellierung ausgehend vom Konzept in der BPA Suite (ARIS Modeller) bis zur ausführbaren Software innerhalb der SOA Suite (JDeveloper).

Aufgrund dieser Durchgängigkeit der Modellierung können grundsätzlich drei verschiedene Vorgehensweisen bei der Modellerstellung angewandt werden:

- Top-down, bereichsorientiert:
Strukturierung des Unternehmens in Funktionsbereiche wie Business Service Areas / Business Support Funktionen
Identifikation relevanter Business Data Objects
- Top-down, geschäftsprozessorientiert:
Modellierung der Geschäftsprozesse/Funktionen mit sog. Capabilities
Ermittlung welche Services als Prozess/Funktionssupport benötigt werden
Modellierung ausgehend von zukünftigen Service Consumern (User/IT-System)
um Requirements der Serviceimplementierung zu klären
- Bottom-up:
Untersuchung existierender Services, Anwendungsfunktionalität und -daten
Beschleunigung von Service Design und Service Orchestrierung durch speziellen BPA Suite Support für Import, Suche und Evaluation existierender Services

Zur Beschreibung der Implementierungsergebnisse des Entwicklungsprojektes wird im Folgenden zuerst die Laufzeitarchitekturplattform der SOA Suite für „human centric BPM“ kurz skizziert. Anschließend werden für die Bereiche Datenobjekte, Services, Human Task Oberflächen und automatische Geschäftsprozesse die Laufzeitartefakte des PSM kurz erklärt und dann für jeden Bereich Modelle und Generatoren für die entsprechenden Laufzeitartefakte untersucht.

Laufzeitarchitektur

Der zentrale Container der Oracle SOA Suite 11g ist eine Service Container Architecture (SCA)-Runtime, die das Deployment von Anwendungen und automatisierten Prozessen in Form von Service-Composites erlaubt.

In den SCA-Container werden auf diesem Weg verschiedenste Servicekomponenten wie z.B. BPEL Engines oder Service-Mediatoren integriert. Die Servicekomponenten aus der SCA-Runtime können

über den Oracle Service Bus (OSB) mit Service Providern und –consumern verknüpft werden. Die Abhängigkeiten und Schnittstellen zwischen den Service-Komponenten werden im SCA-Container mit WSDL und XML-Schema ausgedrückt. Neben dem SCA-Container gehören zur Oracle SOA Suite Laufzeitumgebung noch spezifische Plattformdienstkomponenten wie. z.B. ein Worklist Service, oder ein Notification Service. Die Runtime der SOA Suite und der OSB sind Deployments in den Oracle JEE-5 Server 10.3.1.

Automatisierte Geschäftsprozesse können daher grundlegend mit der SOA Suite als BPEL-Prozesse in einer BPEL-Engine ausgeführt werden. Spezifische Knoten wie z.B. Notifications, Rules oder Human Tasks werden dabei durch Plattformservices ausgeführt. Notifications und Business Rules können in ihrem Verhalten durch entsprechenden Oracle BPEL-Erweiterungen direkt per Deploymentkonfiguration aus dem BPMN- Modell heraus spezifiziert werden. Die Umsetzung von Human Task Oberflächen erfolgt zur Laufzeit über einen Client des Worklist Service. Die SOA Suite besitzt integrierte Werkzeuge zur Erstellung von Human Task-UIs auf Basis der Oracle JSF-Implementierung ADF in der plattform-spezifischen Systemschicht (JDeveloper).

Für die Implementierung eigener Serviceprovider bieten sich EJB3-Komponenten an, da sie durch eine direkte Nutzung der vorhandenen JEE-5 Basisplattform keine neue Systemkomplexität verursachen.

Oracle SOA Suite Runtime Architecture

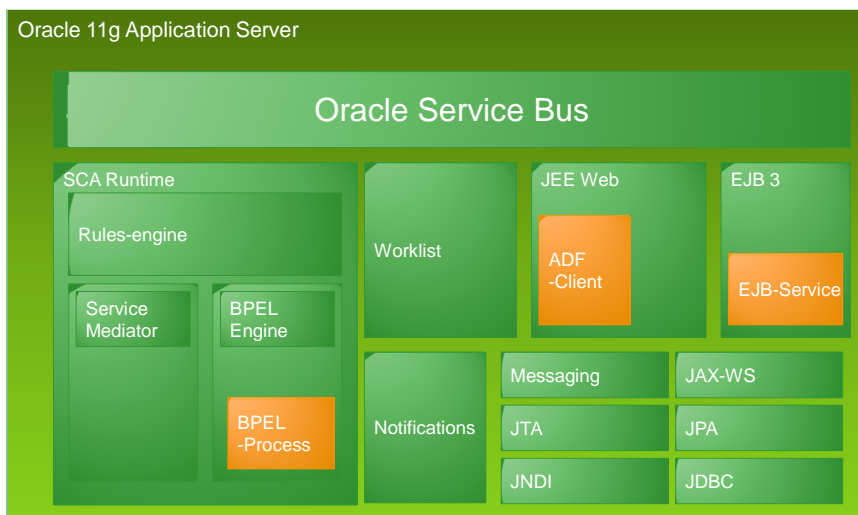


Abb. 1: „Human centric BPM“ SOA Suite Laufzeitarchitektur

Datenmodellierung

Zur Modellierung der benötigten Businessdaten wurde ein bereichsorientierter Topdown-Ansatz gewählt. Dieser Ansatz erfordert die Etablierung eines „Vertrages“ zwischen den beteiligten Anwendungen, die über eine SOA integriert werden sollen. Mit Blick auf die Spezifizierung der Businessdaten führt dieser Vertrag zu einer Kanonisierung der Business Objekte. Kanonische Businessobjekte repräsentieren Businessdaten unternehmensweit einheitlich. Sie unterscheiden sich damit von system- oder anwendungsspezifisch modellierten Repräsentationen von Businessdaten.

Die identifizierten kanonischen Business Objekte Dabei werden als Technical Terms spezifiziert. Die Durchgängigkeit des Modellierungsansatzes erfolgt durch die Verknüpfung der Technical Terms als Ein- bzw. Ausgabedaten in Prozessmodelle höherer Modellschichten.

Technical Terms Modelle gehören der logischen Modellschicht (PIM) an und müssen für eine generative Weiterverwendung mit vertiefenden Modellen erweitert werden. Deshalb werden den Technical Terms technischere Spezifikationen in UML und erweiterten Entity Relation Modellen (eERM) zugeordnet.

Als generierbare Artefakte sind die folgenden Architekturelemente zu diskutieren:

- **Kanonische Dokumenttypen**
Datentypen können in der SOA Suite Laufzeitarchitektur durch XML-Schema repräsentiert werden. Auf diese Typen kann in Serviceschnittstellen, Human Task Formularen, BPEL-Prozesse zurückgegriffen werden.
- **Kanonische Datenobjekte**
Zur Repräsentation der Businesslogik können in der SOA-Suite einfache Java-Objekte genutzt werden, die in der Implementierung von Features wie z.B. Persistenz, Zugriffskontrolle oder Transaktionalität genutzt werden können.
- **Dokumenttypen für Human Task Formulare und Serviceparameter**
Für die Verwendung als Formulare und Serviceparameter sind kanonische Datentypen häufig ungeeignet, da sie anwendungsübergreifend modelliert sind. Der Anwendungscharakter von Formularen und Parametern führt häufig zu Aggregaten aus kanonischen Datentypen, die allerdings häufig nicht vollständig ausgeführt werden müssen.
- **Datenobjekte für Human Task Formulare und Serviceparameter**
Für die Implementierung der Logik von Formularen sind kanonische Datenobjekte aus den gleichen Gründen wie deren entsprechende Dokumente ungeeignet. Auch für die Nutzung als Serviceparameter sind kanonische Parameter nicht in allen Fällen geeignet und müssen daher um spezielle Parameterobjekte erweitert werden.
- **Konverter zwischen XML-Dokumenten und Datenobjekten**
Zwischen den XML Dokumenten und POJOs sind zur Laufzeit Konvertierungen durchzuführen. Diese Konvertierungen erfolgen jeweils an den Prozessgrenzen und sind sowohl für kanonische als auch für Formulare datentypen durchzuführen.
- **Persistenzcode**
Kanonische Datentypen besitzen persistente Attribute und Strukturen. Diese müssen bezüglich ihres Verhaltens in Aspekten wie Ablagestruktur, Multiusersicherheit (Zugriff, Konkurrenz), Konsistenz und Skalierbarkeit implementiert werden. Außerdem ist eine Konvertierung zwischen den persistenten Datenobjekten und deren kanonischen Repräsentationen zu implementieren.

In der ersten Stufe des Entwicklungsprojektes wurde eine generative Nutzung des kanonischen Datenmodells umgesetzt. Dazu wurde als erste Erweiterung der BPA Suite ein Generator implementiert, der aus eERM kanonische Dokumentdatentypen als XSD-Dateien erstellt. Die generierten XSD-Artefakte konnten mit Hilfe der JAXB 2.0 Implementierung als Input für die Generation von kanonischen POJOs und zugeordneten Konvertern genutzt werden. Aufbauend wurde die Modellierung um eERM-Diagramme für Formulare datentypen erweitert und eine passende Erweiterung des proprietären BPA Suite Generators umgesetzt. Diese Modellerweiterungen gingen mit der Modellierung von Human Tasks und Serviceschnittstellen einher bei denen diese eERM für die Modellierung von Formularen und Serviceparametern verwendet wurden. Der Generationsprozess wurde deshalb um Inputartefakte aus der Human Task-Modellierung (Screen Design Model) bzw. Servicemodellierung (Function Allocation Diagram) erweitert, was eine Nutzung der generierten XSDs als Formulare dokumente oder Serviceparameter erlaubte.

Experimentell wurde eine Generierung von Persistenzcode direkt aus den kanonischen Datenmodellen versucht. Dazu sollte aus den eERM eine Transformation innerhalb der logischen Schicht (PIM) in ein Entitypersistenzmodell erfolgen aus dem dann eine Laufzeitrepräsentation in EJB3-Entitäten (PSM) generiert werden sollte. Leider reichte der Input aus den kanonischen eERM nicht für diese Entitytransformation aus, weshalb dieser Versuch abgebrochen wurde. [TODO Gründe: eindeutige Zuordnung, finder, Vernetzung]

Ein Problem der proprietären Generation ist die mangelnde Verknüpfung der generierten Artefakte in die Werkzeuge der Implementierungsschicht (PSM). Die Rückkopplung von generierten XSDs in JDeveloper und den Modellen im Business Process Architect erfolgt über die Verknüpfung des Technical Terms Modells mit einer UML-Repräsentation. Diese Verknüpfung muss für die aus den eERM generierten XSDs eines Technical Terms manuell gepflegt werden. Eine mögliche neue Implementierungsstrategie sollte diese UML-Repräsentation eines Technical Terms stärker berücksichtigen.

Servicemodellierung

Im Rahmen des Top-down Modellierungsansatzes wurden Business Services in Enterprise Architecture Models spezifiziert. Diesen wurden in der logischen Schicht (PIM) Software Service Modelle in Form von Access Diagramms hinzugefügt. Aus den Access Diagrammen können Verweise auf kanonische Businessdaten und Parameter über die entsprechenden Technical Terms modelliert werden. Als Ziel für eine generative Weiterverwendung der Access Diagramme bieten sich die Schnittstellendefinitionen der Services in Web Service Description Language(WSDL) an. Generate für die eigentliche Implementierung der Service Provider bzw. Consumer wurden im Laufe des Projektes nicht realisiert. Aus den WSDL-Artefakten kann stattdessen die Unterstützung der Oracle SOA-Suite für die Realisierung von Service Providern und Service Consumern genutzt werden. Die Vorgehensweise des Werkzeugeinsatzes entspricht dabei einem „Contract First“-Implementierungsansatz.

Die BPA-Suite wurde deshalb um einen Generator für WSDLs erweitert. Dieser Generator nutzt Access-Diagramme als primären Input. Für die Generation der Messagetypen die in den Operationen der Serviceschnittstellen benutzt werden, greift der Generator auf die verknüpften Technical Terms Modelle zurück. Dadurch kann in den WSDL eine Wiederverwendung von kanonischen Datentypen, als auch eine Modellierung spezieller Parameterobjekte aus XML-Schemageneraten der Datenmodellierung erfolgen.

Grundsätzlich können alle beteiligten Services mit Ihren Schnittstellen über diesen Ansatz modelliert und generiert werden. Die Modellierung und Generierung erfolgte gemäß dem Top-Down Ansatz für die Schnittstellen der Prozessservices und der aus Business Services abgeleiteten Domain Services.

Ansatzbedingt kam es in der technischen Implementierung der Services zur Notwendigkeit der Verfeinerung von Serviceschnittstellen insbesondere mit Blick auf die Struktur von Parametern bzw. die Optimierung des Einsatzes von tiefer strukturierten kanonischen Datentypen. Bestimmte Schnittstellenaspekte wie z.B. Feldvalidierungen und Parameterstrukturen aber auch Fehlerverhalten werden aus der Erfahrung des Projektes in einem Top-down Ansatz nur ungenügend für eine Generation spezifiziert. Ein weiterer Faktor der zu Anpassungen der Schnittstellen führte war die Frage nach der performanten Verfügbarkeit von Businessdaten. Analytisch wurden häufig Referenzen auf kanonische Datentypen in Schnittstellen modelliert, deren Verfügbarkeit für den Serviceprovider nicht ausreichend untersucht wurde. Diese zu starke Fokussierung auf die Sicht des Servicekonsumenten in der Analyse ist vermutlich ein grundsätzliches Problem eines Top-down Vorgehens. In späteren Projektphasen und Folgeprojekten soll bereits in der Analysephase stärker auf solche Implementierungsaspekte in der Modellierung der Access Diagramme und eERM geachtet werden.

Als technische Schwierigkeit beim beschriebenen Vorgehen erwiesen sich die bereits bei den entsprechenden Datenmodellen erklärten Probleme mit der mangelnden Integration der XSD-Generat

in JDeveloper. Neben der manuellen Pflege der verknüpften UML-Repräsentationen der Technical Terms, verhinderte die Generation der WSDLs auch ein echtes Reverse Engineering der WSDLs in der plattformspezifischen Schicht (PSM). Eine einfache Anpassung der WSDL-Generatoren in JDeveloper um die festgestellten Implementierungsmängel der WSDLs zu beheben war daher nicht möglich.

Die Korrektur der Schnittstellen musste immer modellgetrieben über die Access Diagramme und deren Verknüpfungen erfolgen. Diese Arbeitsweise erwies sich wegen ihres mittelbaren Charakters im Projektverlauf als suboptimal und soll zukünftig durch weitere Anpassungen am Servicegenerator vermieden werden.

Human Task Modelle

Die Interaktion von Benutzern mit automatisierten Geschäftsprozessen wird in den BPMN-Prozessmodellen der BPA Suite durch produktspezifische Erweiterungen des BPMN-Standards abgebildet. Das entsprechende Syntaxelement ist ein Human Task Knoten. Neben der logischen Flußdeklaration einer Benutzerinteraktion können weitere beschreibende Attribute wie z.B. Benutzerrolle, Interaktionsmuster oder Taskdokumente im Modell am Knoten spezifiziert werden.

In Vorstudien zum Projekt wurde mit Möglichkeiten der interaktionsmusterbasierten UI-Generation aus den Ein- bzw. Ausgabedokumenten eines Tasks experimentiert. [TODO Verweis auf Naked Objects] Insbesondere die Benutzbarkeit der Oberflächen entsprach nicht den geforderten Qualitätsmerkmalen.

Für die Generierung einer qualitativ angemessenen Implementierung des Userinterfaces derart deklariert Benutzerinteraktionen reichen die Informationen im BPMN-Modell allein nicht aus. Unter anderem fehlen im BPMN-Modell die folgenden notwendigen Informationen:

- Spezifikation der Aus- und Eingabeformulare
Ziel der Modellierung von speziellen Formularen ist deren Optimierung mit Blick auf die kognitiven Möglichkeiten des Benutzers. Im Gegensatz zu generischen Formularen, welche lediglich Zugriff auf die Inhalte von Dokumenten über deren Schemastrukturen bieten, wurde die Möglichkeit der freien Aggregation, Ausblendung und Anordnung von Inhalten über Dokumentenschemata hinweg benötigt. Insbesondere die Verwendung von kanonischen Dokumenten in der Analyse braucht diese Möglichkeiten den Benutzer von der Systemkomplexität abzuschirmen.
- Spezifikation des logischen Flusses zwischen Formularen
Um die Benutzerführung bei komplexen Ein- bzw. Ausgabedokumenten weiter zu erleichtern, sollte eine Zerlegung in logische Flüsse von kleineren Formularen erfolgen. Größe und Umfang der Formulare orientieren dabei sich an den kognitiven Möglichkeiten der Benutzer und nicht an der logischen Struktur der Businessobjekte bzw. Dokumente. Gleichzeitig kann mit diesen kleineren Formularen versucht werden, Wiederverwendungspunkte im analytischen Modell zu ermöglichen.
- Spezifikation von Formularverhalten
Eine Besonderheit moderner Formularanwendungen ist die Verwendung komplexer Viewcontrols, wie z.B. aktive Tabellen oder Bäume. Die Einbindung dieser Visualisierungskomponenten erleichtert den Benutzern die Bearbeitung komplexer Datenbestände. Die UI-Modelle können über reine Verwendungsdeklarationen solcher komplexer Komponenten verfügen oder darüber hinaus sogar die Spezifikationen von konkreten Verhaltenserweiterungen ermöglichen.

In der Top-down Analyse des Systems wurden zu diesem Zweck zwei weitere Diagrammtypen innerhalb der logischen Schicht(PIM) modelliert. Die Spezifikation der Formulare und ihres Verhaltens erfolgte mit Screen Design Diagrammen. Diese enthalten Syntaxelemente zur Beschreibung des Layouts und der Beziehungen zwischen den Elementen der Taskdokumente und den

Formularelementen. Die Spezifikation des Formularverhaltens wird durch deklarative Syntaxelemente für komplexe Controls, sowie die Stereotypisierung von Formularen unterstützt. Dazu wurden aus den Vorstudien des Projekts verschiedene Formularstereotypen als sogenannte Templates für die Screen Design Diagramme erstellt. Mit Screen Navigation Diagrammen wurden zusätzlich die logischen Flüsse von Formularen modelliert. Die Screen Navigationen konnten dann in der BPA Suite mit den Human Task Knoten der BPMN-Diagramme verknüpft werden.

In der generativen Umsetzung der UI-Modelle wurde eine Abwandlung der Standardarchitektur der Oracle SOA Suite vorgenommen. Als Clientplattform entschied sich das Projektteam gegen die bestens in der SOA Suite integrierte JSF-basierte Umsetzung für eine auf Microsoft Silverlight basierte Strategie. Ursächlich für diese Entscheidung waren ausschließlich nicht technische, nicht funktionale Requirements wie z.B. der Support für den Microsoft Internet Explorer 6. Als generierbare Artefakte für diese Plattform wurden die Oberflächendeklarationen in XAML identifiziert. Diese konnten direkt aus den stereotypisierten Screen Diagrammen generiert und verwendet werden, was insbesondere in den Analyseworkshops mit Endanwendern die Akzeptanz des Ansatzes deutlich erhöhte. Um die Screen Navigation Diagramm generativ ausnutzen zu können, wurden verschiedene Optionen diskutiert. Neben der direkten Generation von C#-Code zur Flusststeuerung im Client, wurden server- bzw. clientseitige Interpreter für unterschiedliche Flow-Control-Sprachen evaluiert. Unter Abwägung der Gesamtkosten der Realisierung wurde schließlich eine proprietäre XML-Flußsprache gewählt. Eine dazu passende Generatorimplementierung sowie die Realisierung eines zustandsbehafteten serverseitigen Interpreters sollen die unternehmensweite UI-Generationsstrategie vervollständigen.

Die auf diesem Weg aus den analytischen Modellen erhaltenen Userinterfaces erreichten trotz deutlicher Verbesserungen gegenüber den Vorstudien nicht die gewünschte Benutzerakzeptanz und wurden deshalb manuell angepasst. Da ein mögliches Roundtripengineering der analytischen Modelle und ihrer Generate bereits wegen der Probleme mit der Servicemodellierung nicht durchgeführt wurde, verursachte diese Nachbearbeitung keine zusätzlichen Projektkosten.

Durch gezielte Anpassungen der Screen Navigation und Screen Design Modelle mit Blick auf die Verwendung spezieller Formulardokumente und taskspezifischer Navigationen lassen sich die Generate zwar deutlich verbessern, allerdings steigt der Aufwand der Modellierung überproportional an. Insbesondere der Einsatz der UI-Modellierung in Analyseworkshops wird deshalb im Projektunternehmen derzeit überarbeitet.

Prozess Modelle

Die konzeptionelle Modellierung automatisierter Geschäftsprozesse wurde in der BPA Suite mit erweiterten Ereignis-Prozess-Ketten (EPK) modelliert. In der logischen Modellierung (PIM) bietet die BPA Suite EPC- und BPMN-Modelle und deren Kombination an. BPMN-Modelle können in der Oracle 11g SOA Suite direkt ausgeführt werden (WYSIWYE). Alternativ kann vor der Installation in die SOA Suite eine Transformation von BPMN in BPEL durchgeführt werden. Die BPEL-Modelle stellen dabei die physikalische Modellschicht (PSM) dar. Diese Transformation der BPMN-Modelle vor ihrer Ausführung ist das etablierte Modellierungsvorgehen im Projekt, da es bereits in einer Projektvorstudie mit der Oracle SOA Suite 10 erfolgreich umgesetzt wurde. Eine Besonderheit dieser Modellierung ist die Möglichkeit spezifische BPMN-Syntaxerweiterungen der BPA Suite wie z.B. Human Tasks und Automated Activities direkt in BPMN-Modellen verwenden zu können. Da die BPMN-Modelle also entweder direkt oder durch Transformation in BPEL ausgeführt werden können ist eine weitere generative Aufbereitung der Modelle unnötig. Im Sinne eines MDSD-Ansatzes für BPMN ist die Oracle BPA Suite daher als ausreichend instrumentiert zu betrachten.

Der gewählte Top-down Ansatz der Modellierung über EPK verursachte allerdings Qualitätsprobleme der auf diesem Weg erstellten BPMN Modelle. Da die initiale Erfassung der Requirements in EPC-Modellen erfolgte, wurde ein eindeutiges Tracking aller EPC-Modellelemente auf BPMN-Modellelemente aus Gründen des Requirementsmanagements nötig. Obwohl in der Literatur

automatisierte Transformationen von EPC in BPMN beschrieben sind, wurde aufgrund fehlender Umsetzbarkeit in der BPA Suite eine manuelle Umwandlung durchgeführt. Die aus den EPC transformierten BPMN-Diagramme mussten für eine direkte Ausführbarkeit anschließend weiter angereichert werden. Aus den EPC konnten z.B. keine technisch ausreichend detaillierten Informationen über die im Prozess benutzten Operationen der Software Services entnommen werden und mussten deshalb nachträglich modelliert werden. Der Transformationsschritt offenbarte weitere Probleme der BPMN-Modelle. Für die sinnvolle Ausführbarkeit von BPEL ist die Deklaration von Prozesspayloaddokumenten und -variablen notwendig. Diese Deklarationen mussten ebenfalls nachträglich modelliert werden, da sie nicht durch eine Transformation aus EPC entnommen werden konnten. Auch eine automatisierte Transformation (Generierung) der BPMN-Modelle aus den EPCs dürfte diese Probleme nicht umgehen können. Kostenseitig muss auch hier wieder die Möglichkeit einer Rückkopplung von plattformspezifischen Modellen (BPEL) in JDeveloper und logischen Modell im BPA-Architect berücksichtigt werden. Mit Hilfe dieses Reverse Engineerings lassen sich die Mängel im BPMN-Modell kostengünstig direkt in der Implementierungsschicht beheben.

Fazit

Die Oracle 11g SOA Suite liefert sowohl eine sofort einsatzfähige Basis für eine modellgetriebene SOA als auch eine Laufzeitarchitekturplattform für modellgetriebene „human centric“ BPM. Die Integration der verschiedensten im Projekt genutzten Softwarekomponenten und Werkzeuge der Oracle 11g, die von der SOA Suite über den OSB bis in die BPA Suite reichte war ein Erfolgsfaktor des Projektes. Dabei lässt die Implementierungsqualität einzelner Features jedoch Raum für Verbesserungen. Mit Blick auf die Möglichkeit der generativen Softwareentwicklung konnte die BPA Suite besonders durch ihre umfangreichen Modellierungssprachensupport (z.B. ARIS, UML) und ihre leichte Erweiterbarkeit überzeugen.

Der im Projekt gewählte Top-down Ansatz der Serviceevaluierung, der von EPC/ARIS Modellen ausgehend zur BPMN Modellierung eingesetzt wurde, integriert sich leider nicht vollständig in die BPA Suite Werkzeugkette, da für seine Durchführung Modellredundanzen akzeptiert werden mussten. Insbesondere die im Projekt implementierten Generatoren bieten keine Möglichkeit des Reengineerings zwischen PSM und PIM, was zu Redundanzen bei WSDL- und Schemamodellen (ARIS/UML) führte. Die kritische Betrachtung der Kostenseite zeigt daher auf, dass nur bei hohen initialen Modellierungsanstrengungen ökonomisch interessante Ergebnisse erreicht werden können. Ein nachträgliches Verbessern oder Anreichern der Modelle ist wegen der mangelnden vollständigen Roundtrip-Fähigkeit der Werkzeuge kaufmännisch nicht empfehlenswert.

Nach erfolgreicher initialer Projektumsetzung soll zukünftig im Projektunternehmen das beschriebene Vorgehen weiter angewendet und sowohl in den Analyseverfahren als auch in den generativen Aspekten weiter verbessert werden.

Kontaktadresse:

Name

Orientation in Objects
Weinheimer Strasse 68
D-68309 Mannheim

Telefon: +49 (0) 621-718 390
Fax: +49 (0) 621-718 3950
E-Mail christian.dedek@oio.de
Internet: www.oio.de