

# **Kanonische (XML-)Datenmodelle für die Oracle SOA Suite**

**Marcel Amende  
ORACLE Deutschland B.V. & Co. KG  
Düsseldorf**

## **Schlüsselworte:**

Kanonisches Datenmodell, SOA Suite 11g, Service Bus (OSB), Enterprise Repository (OER), Service Registry (OSR), Application Integration Architecture (AIA), GS1, EPCGlobal, XML

## **Einleitung**

Man muss das Rad nicht immer neu erfinden: AIA (Foundation Pack), OAGi, GS1 XML & co. bieten allesamt Branchen- und Themenorientierte XML Schemata, die sich über den eCommerce hinaus für die Verwendung als vereinheitlichtes Datenmodell in der Oracle SOA Suite anbieten. Konzept, Nutzen und Einsatzbereiche von einheitlichen Datenmodellen in einer SOA werden diskutiert, die praktische Anwendung innerhalb der Oracle SOA Suite - vom Repository, über den JDeveloper bis hin zu BPEL und Service Bus - demonstriert.

## **Kononische Datenmodelle**

Die Entwicklung von Systemen erfolgt meist unabhängig voneinander. Ein Unternehmen wird einen Teil seiner Applikationen bei verschiedenen Anbietern vorgefertigt einkaufen, andere im Haus selbst entwickeln. Die jeweiligen Entwickler werden bei ihrer Arbeit wohl nur die jeweils eigene Applikation bedenken. Dies führt dazu, dass jede Applikation auf ihre eigenen Daten- und Nachrichtenformate aufsetzt. Bringt man diese Applikationen dann in seinem Unternehmen zusammen, um durchgängige Geschäftsprozesse abwickeln zu können, steht man einem wahrhaft „Babylonischem Sprachwirrwar“ gegenüber.

Mittels eines Enterprise Service Bus (ESB) lassen sich Systeme komfortabel und deklarativ, also ohne Programmieraufwand, miteinander verbinden: Adaptoren bedienen die verschiedensten Transportkanäle und übersetzen die übermittelten Daten in XML-Dokumente, so dass im Bus die XML-Struktur des Quellsystems nur noch in die Zielstruktur transformiert werden muss. Für eine begrenzte Anzahl an Systemen ist diese Vorgehen ausreichend, mit wachsender Zahl beteiligter Systeme wächst so aber der Bedarf an Transformationen exponentiell an. Geht man davon aus, dass bei der Kommunikation zwischen zwei Systemen für den Hin- und Rückweg jeweils eine Transformation benötigt wird, errechnet sich für die Punkt-zu-Punkt-Anbindung von 8 Systemen die unglaubliche Zahl von 56 Transformationen, wahrhaft unüberschaubar! Eine einzige Änderung an der Schnittstelle eines der acht Systeme würde 14 Änderungen in Transformationsheets verursachen.

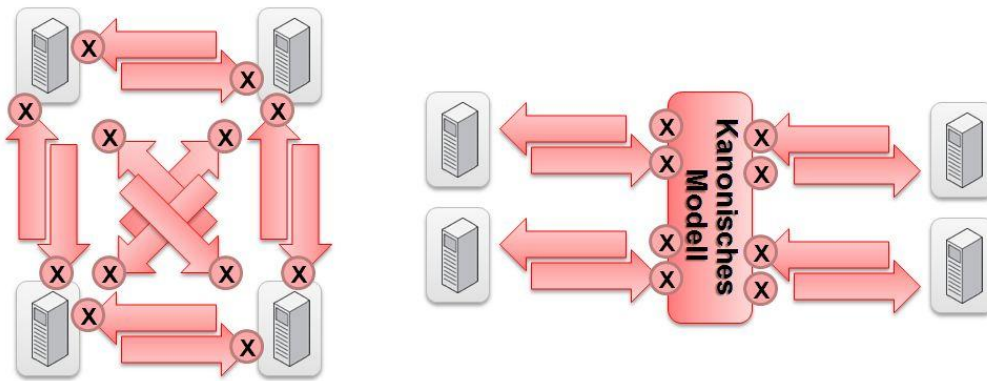


Abb. 1: Vergleich von Punkt-zu-Punkt und kanonischen Verbindungen zwischen vier Systemen

Alternativ dazu kann man ein einheitliches (kanonisches) Datenmodell einführen, in das die Nachrichteninhalte aller Applikationen transformiert werden und das dann ohne weitere Transformation von einer Applikation an die andere weitergeleitet werden kann. Ein Applikation braucht dann keinerlei Kenntnis mehr von den Datenformaten der anderen Applikationen. Bei einer Änderung an einer Schnittstelle eines Systems müssen nur noch die beiden Transformationen in das kanonische Datenmodell angepasst werden. Zwischen acht Systemen wären damit nur noch insgesamt 16 (vormals 56!) Transformationen notwendig.

Anzahl Systeme	Anzahl der Transformationen, Punkt-zu-Punkt Verbindung	Anzahl der Transformationen, einheitliches Datenmodell
2	2	4
3	6	6
4	12	8
5	20	10
6	30	12
7	42	14
8	56	16

Tab. 1: Anzahl von Transformationen für Punkt-zu-Punkt und kanonischen Verbindungen

### Kanonisches Datenmodell

Entscheidet man sich nun für die Einführung eines kanonischen Datenmodelles, um zu einer wahrhaft losen Kopplung zwischen den Systemen zu kommen, schliessen sich schnell elementare Fragen nach Konzeption, Erstellung, Pflege, Verwaltung und Einsatz eines solchen Modells an.

Beschäftigen wir uns zunächst mit inhaltlichen Aspekten: Bei Webservice-basierten Systemen sind die Datenformate schon innerhalb der WSDL als XML-Dokumente beschrieben. Der Aufbau der XML-Dokumente wiederum wird meist über eine „XML Schema Definition“ (XSD) definiert und validiert. Diese XSDs dienen konsequenterweise auch zur Variablendefinition in BPEL, BPM und OSB, können in diesen Werkzeugen somit direkt verwendet werden. Dafür importiert man die XSDs in die jeweilige Entwicklungsumgebung oder nutzt die Integrationsmöglichkeiten vom JDeveloper mit dem „Oracle Enterprise Repository“, um die XSDs aus einer zentralen, strukturierten, versionierbaren und kollaborativen Ablage zu beziehen.

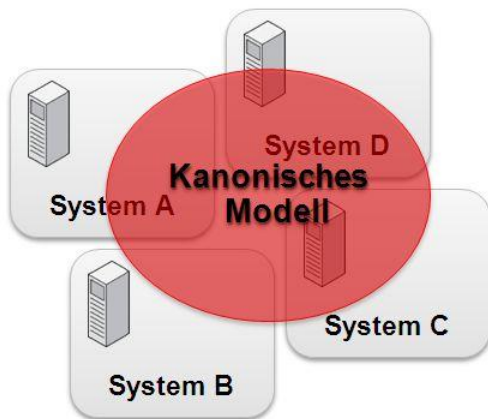


Abb. 2: Kanonisches Modell als Obermenge der schnittstellenrelevanten Applikationsdaten

Um applikationsübergreifend Daten austauschen zu können, wird das kanonische Modell eine Art Obermenge der Datenmodelle aller beteiligten Applikationen bilden. Glücklicherweise sind naturgemäß bei weitem nicht alle von den Applikationen intern verwendeten Daten relevant für den Austausch über externe Schnittstellen. Demnach besteht die Herausforderung darin, ein gleichzeitig kompaktes und funktional vollständiges Datenmodell zu erstellen, da Änderungen am kanonischen Modell schnell Änderungen an allen Transformationen zur Folge haben können. Diese Herausforderung kann man sich selbst stellen, es ist aber eine Überlegung wert, Erfahrungen von Branchen- und Softwareexperten einzubeziehen und auf bestehende, marktetablierte Datenmodelle aufzubauen.

### **Application Integration Architecture (AIA)**

Die weit grösste praktische Erfahrung in der Integration von Unternehmenssoftware wird man bei Oracle selbst finden. Hier wurde die strategische Entscheidung getroffen, die eigene Unternehmenssoftware auf Basis einer Service-orientierten Architektur zu entwickeln. Auch die Integration zwischen einzelnen Produkten - sowohl aus der eigenen Schmelze, als auch von anderen Herstellern - fusst auf diesem Ansatz. Vorgefertigte Integrationsprozesse hierfür sind als „Process Integration Packs“ (PIP) erhältlich. Als Basis für eigene Integrationslösungen kann das „Foundation Pack“ herangezogen werden. Dieses beinhaltet u.a. ein SOA Datenmodell in Form sogenannter „Enterprise Business Objects“ (EBO), die Geschäftsdatenobjekte für z.B. Kunden- und Bestelldaten als XSD bereitstellen. Alle zum Foundation Pack gehörenden Daten-, Nachrichten- und Schnittstellenformate werden als fertiges Service Repository geliefert, um Sichtbarkeit, Wiederverwendung und Analysefähigkeit der Objekte im gesamten Unternehmen und über den gesamten Lebenszyklus, von der Entwicklung bis zum Betrieb, zu gewährleisten.

### **OAGi**

Einen ähnlichen Ansatz verfolgt die „Open Applications Group“ (OAGi), die als non-profit-Organisation die Erstellung prozessbasierter Standards für die Unternehmensintegration, den elektronischen Datenaustausch und das Cloud Computing vorantreibt. Durch die Tatsache, dass dies branchenübergreifend betrieben wird, ist hier eine mächtige Library an Objekten, Services und Prozessen erwachsen, die in Form eines Archivs von XSDs und WSDLs zur Verfügung gestellt wird.

Die Artefaktsammlung ist deutlich entwicklungsorientiert, d.h. gemeinsam genutzte Elemente werden in den z.B. branchenspezifischen XSDs importiert, um den Pflege- und Erstellungsaufwand für die Community gering zu halten. Dies führt aber dazu, dass die Artefakte aus Performancegründen weder

in den Entwicklungsumgebungen, noch in den Laufzeitumgebungen direkt und ohne Anpassungen verwendet werden können. Hier ist zunächst eine Filterung auf relevante (Branchen-) Ausschnitte und das Flachklopfen (reduzierung der „imports“) der XSDs nötig. Hierfür stellt OAGi seinen Mitgliedern spezielle Werkzeuge bereit.

### **Business-to-Business und eCommerce**

Auch moderne B2B-Standards werden unter Verwendung von XML und unter Anwendung von SOA-Konzepten definiert. Dies zeigt sich insbesondere in technologisch neuen Themenbereichen. In den Standarddefinitionen von EPCGlobal beispielsweise, originär geschaffen für die Verarbeitung von Informationen aus automatischen Datenerfassungssystemen (RFID) und den Aufbau eines globalen Netzwerkes für den Austausch serialisierter Artikelinformationen („Internet der Dinge“), finden sich XML-Datenmodelle für die Beschreibung einzelner Handgriffe z.B. im Umgang mit Produkten und Transportmitteln. Diese lassen sich hervorragend für den Aufbau eines Herkunfts- und Produktverfolgungssystems („Track & Trace“) nutzen. Auch die klassische B2B-Kommunikation folgt langsam diesem Trend. Die Standardisierungsorganisation GS1 bietet mittlerweile XML-basierte Alternativen zum herkömmlichen UN/EDIFACT an, die sich transformationsfrei direkt in SOA-Infrastrukturen nutzen lassen.

### **Kontaktadresse:**

#### **Marcel Amende**

ORACLE Deutschland B.V. & Co. KG  
Hamborner Str. 51  
D-40472 Düsseldorf

Telefon: +49 (0) 211-74839 539  
Fax: +49 (0) 211-74839 222  
E-Mail: [Marcel.Amende@oracle.com](mailto:Marcel.Amende@oracle.com)  
Internet: [www.oracle.com](http://www.oracle.com)