

SCA - Service Component Architecture

Torsten Winterberg (Opitz), Berthold Maier (Oracle)

Schlüsselworte: SOA, Architektur, SCA

Einleitung

Service Component Architecture (SCA) ist das von der OAGIS standardisierte Komponentenmodell zur Entwicklung von Anwendung auf Grundlage des serviceorientierten Architekturgedanken (SOA). Primäres Ziel ist ein einheitliches und technologieunabhängiges Komponentenmodell auf den Eckpfeilern der SOA-Standards zu schaffen. Die sogenannte SCA-Composites stellen in dem Modell die Komponentenkapsel dar und fungieren in ihrer Außenansicht als normale WebService. Dabei ist es das SOAP Protokoll nicht zwingend und die Elemente in einem SCA können durch ganz normale Java Klassen (POJO's), Spring Komponenten oder EJB's, aufgebaut sein. SCA-Composites werden mit der Schnittstellendefinitionssprache WSDL beschrieben und können als Webservice ohne weiteren Programmierung der öffentlichen Service-Welt zugänglich gemacht werden.

Ein wesentliches Ziel der Spezifikation ist u.a. das gesamte Technologie-Spektrum von heute und der Zukunft abzudecken, weshalb die SCA-Konzepte aufgeteilt und die Technologieanbindungen als separate Spezifikation (SCA-Binding) eingebracht werden. Das Hinzufügen einer Zukunftstechnologie und das Mischen unterschiedlicher Technologiekomponenten wie BPEL, Java, Spring, PL/SQL, ist somit heute und zukünftig möglich. Die geforderte Interoperabilität und Zukunftsfähigkeit wird mit diesem Konzept best möglich gewährleistet.

Auswirkung auf die Anwendungsarchitektur

Auf SCA Basis ist es nun einfache möglich ein unternehmensweites Servicemodell in einer bestehenden Anwendungslandschaft – auch Java, Spring, JEE/EJB, C#, und C++ usw. - zu etablieren. Damit das Serviceparadigma in den Betriebe Einzug hält und einen wirklichen Businessmehrwert liefert (ROI), müssen von heute an die Architekten und Programmierer beim Anwendungsdesign umdenken.

Bestehende Anwendungen sind beim Umsetzen des serviceorientierten Gedankens mit SCA zu zerhacken und sollten nur noch aus Einzelbausteine (Service) bestehen. Das Zusammenfügen der Programmbausteine (Service-Komponenten) zu höherwertigen Komponenten und dessen Zusammenspiel sollte in den SCA-Composites durch Einbindung der SCA-Servicekomponenten wie Java, Spring, EJB, PL/SQL-Module usw. geschehen und nicht mehr im Anwendungscode selbst. Service werden mit diesem Modell loser gekoppelt und Ablaufänderungen bzw. Prozessänderungen lassen sich leichter - visualisiert mit SCA Editoren - umsetzen.

SCA und Oracle Fusion

Oracle hat sich schon vor Jahren entschieden, die Fusion Plattform auf Basis einer serviceorientierte Architekturen mit SCA Komponenten umzusetzen.

Durch die konsequente Trennung von Technologie und Serviceschnittstelle verspricht sich Oracle von SCA ein langfristiges stabiles Architekturumfeld zu schaffen und eine strategische langlebige Plattform für Oracle Fusion CRM (Siebel), Applications (ERP), PeopleSoft usw. zu schaffen – das war ja leider mit JEE etwas verloren gegangen.

Mit dem SOA Suite Release 11g wurde die Infrastruktur dazu fertig gestellt und die bestehende Produkte BPEL, Mediator, Rules als SCA-Komponenten ausgeprägt. Konsequenterweise müssen nun alle 11g Programme wie BPEL, HW, Rules usw. migriert und als SCA-Service der Außenwelt zur Verfügung gestellt werden. Dazu wurde das Migrationstool SmartUpdate – ist u.a. im JDeveloper

integriert – entwickelt, das die vorhandenen BPEL Prozesse automatisch in SCA-Komponenten überführt. Bei einer größeren Migration sollte allerdings das Standalone Tool verwendet werden, da sich das Service-Programmiermodell von 10g nun 11g völlig geändert hat und in der Praxis meist verschiedene Komponenten (BPEL Prozesse) zu einem SCA-Projekt zusammengefasst werden müssen, was damit einfacher möglich ist.

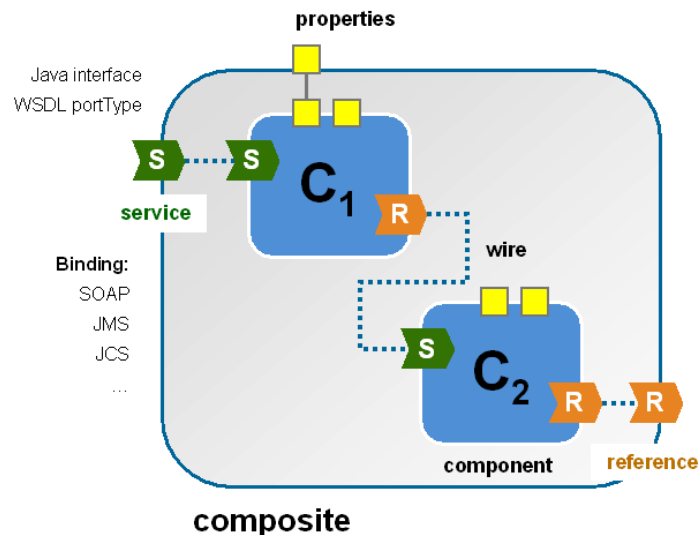


Abb. 1: SCA - Schematische Darstellung

SCA Grundkonzepte

Composite / Modul

Das Composite stellt innerhalb der SCA ein Modul von eingebetteten Komponenten, einer Schnittstelle und gehaltenen Verbindungen nach außen und innen dar. Das SCA-Design erlaubt ein sogenanntes SCA-Composites lokal oder entfernt zu betreiben. Die Kommunikation muss dabei nicht zwingend SOAP sein, sondern kann auch JMS, JavaCall oder eine RMI Kommunikation nutzen. Die Verteilung und Konfiguration der Komponenten, sowie die Einstellung des verwendeten Protokolls wird beim Deployment, also nach der SCA Entwicklung festgelegt und kann jederzeit an die geforderte Systemarchitektur angepasst werden.

Die SCA Architektur erlaubt somit eine vollständige Anwendung durch ein SCA-Composite zu bündeln und die angebotene Funktionalität durch das verwenden von SCA-Komponenten und anderen SCA-Composites bzw. externen Webservice zusammenzufügen. Das Bündeln der Komponenten zu einem Composites erfolgt immer in der einzelnen XML-Datei composite.xml.

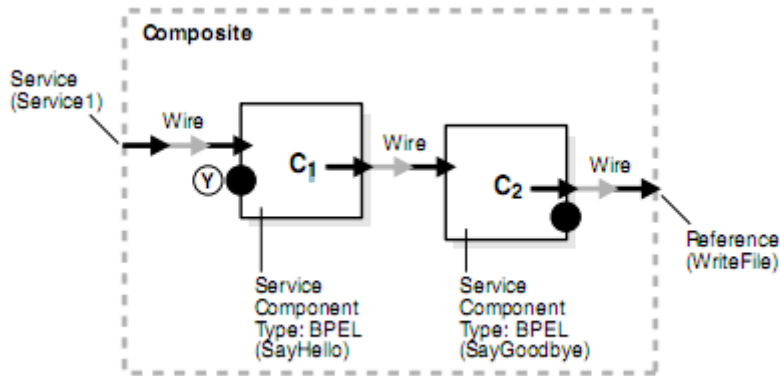


Abb. 2: SCA - Schematische Darstellung

EntryPoints

Einstiegspunkte (EntryPoints) stellen in einem SCA-Composite die nach außen angebotenen Dienste (Service) mit einer definierten Schnittstelle (WSDL-Operationen) dar. Die Namen der EntryPoints müssen in dem SCA-Modul eindeutig sein.

Components

SCA unterscheidet zwischen Komponentenarten (Component Type) und dessen Verwendung. Beispiel für Komponentenarten sind BPEL, RuleEngine, Mediator, HumanTask, SpringBean, usw. Jede Komponente kann beim Einbetten in ein SCA-Composite eine lokale Konfiguration erhalten und stellt seine Dienste in Form einer WSDL zur Verfügung.

Referenzen

Referenzen wurden zum Einbinden externer Dienste (Service) geschaffen, die intern von der SCA-Komponente genutzt werden, um dessen Funktionalität durch Nutzung dieser Service zu erweitern. Referenzierte Dienste werden stets außerhalb des lokalen SCA-Ökosystems verwaltet. Bei referenzierten Dienste kann es sich um externe SCA-Komponenten, JCA-Adapter oder gar um einen einfachen Webservice handeln. Sie müssen lediglich dem bereitgestellten Webservice-Interfaces WSDL gehorchen.

Wires

SCA-Komponenten können in einem SCA-Composite mittels sogenannter Wire mit anderen SCA-Komponenten verbunden werden.

Wires sind rein virtueller Natur und dienen der Kommunikation zwischen Service-Referenzen. Eine erfolgreiche Kommunikation kann nur aufgebaut werden, wenn Service-Anbieter und Service Konsument über die gleiche Schnittstelle (WSDL und PortType, Java Interface, EJB) verfügen. Mittels Bindingangabe kann ein Service-Anbieter angeben, welches Kommunikationsprotokoll (SOAP, JMS, RMI oder JavaBean) er zur Verfügung stellt.

Service Data Objects

SDO ist eine weitere Spezifikation im SCA-Umfeld, die zur Realisierung von DatenServices verwendet werden sollte. SDO Datenobjekte sind per XSD definiert und werden als XML-Datengraphen zwischen den einzelnen Komponenten ausgetauscht. Sie stellen einen Arbeitsausschnitt einer Datenquelle dar. Änderungen und damit die Zurückführung bzw. Update der zugrundeliegenden Datenquelle (meist Datenbank) wird durch das ChangeSet vom SDO selbst verwaltet. Aufwändige SOA CRUD Service (Create-, Update-, Delete- Operationen) entfallen damit weitgehend und man

kann sich in der SOA auf die fachliche Service konzentrieren. Realisiert wird die SDO Persistenz-Abbildung innerhalb der Oracle SOA Suite durch das EclipseLink Framework (ehem. Toplink). SDOs können in der Architektur gekoppelt oder entkoppelt arbeiten und abhängig davon transaktionsgebunden mit den Service-Komponenten eines Composites laufen oder nahgelagert in einer gesonderten Transaktion abgearbeitet werden. Dieser Aspekt hat starke Auswirkungen auf das Design und sollte stets mit berücksichtigt werden.

Innerhalb des Oracle Fusion Stack's sind die ADF-ViewObjekte auch direkt als SDO mit WSDL-Schnittstelle exponierbar. Damit lässt sich ein einmal in ADF programmierter Logikbaustein durch die SOA, WEB-Anwendungen oder Java aufrufen und die maximale Wiederverwendung ist gewährleistet.

Die Seitenzahl wird von uns eingefügt!

Bitte fügen Sie Ihre Kontaktadresse hinzu.

Kontaktadresse:

Berthold Maier

Oracle Deutschland
Riesstraße 25
D-80992 München

Telefon: +49 89 1430 2788
E-Mail berthold.maier@oracle.com
Internet: www.oracle.com

Torsten Winterberg

OPITZ CONSULTING GmbH
Kirchstr. 6
D-51647 Gummersbach

Telefon: +49(0)2261-6001-0
E-Mail torsten.winterberg@opitz-consulting.de
Internet: <http://www.opitz-consulting.de>