

Werkzeuge der Service Orientierten Software Entwicklung

Markus Nöbauer

MatNr. 0155848 SKZ: 880

<http://tools4SOSE.lin2win.net>

Abstract

Durch den Internet Boom und der Verfügbarkeit von offenen, akzeptierten Standards (WSDL, SOAP, usw) hat das Thema Service Orientierte Architekturen (SOA) rasch an Bedeutung gewonnen. Alle grossen Hersteller von Unternehmenssoftware wie BEA, IBM, SAP, SUN uvm. haben in irgend einer Form SOA Anwendungen und Tools in ihrem Portfolio. Die meisten dieser Softwarehäuser bieten nicht nur fertige oder halb-fertige, anpassbare Unternehmenssoftware an, sondern auch die Werkzeuge um SOAs zu implementieren. Das SOA Paradigma betrifft den ganze Bereich von Unternehmenssoftware. Somit sind auch alle Tools aus diesem Bereich betroffen. Sie müssen die SOA Kernkonzepte unterstützen, nachrüsten oder durch neue Tools ersetzt werden. Im Wesentlichen lassen sich 4 Kernbereiche identifizieren; Modellierung eines Business Designs und dessen transformation in eine Orchestrierung von Services. Die Implementierung und Nutzung von Services, vorzugsweise Webservices basierend auf offenen Standards. Hosten, Wartung und Monitoring von Services auf einem Enterprise Service Bus. Auswertung der laufenden SOA und Rückschlüsse auf das Business Design.

1. Introduction

Es würde den Umfang der Arbeit sprengen alle Anbieter, Tools und Lösungen genauer vorzustellen. Auf den folgenden Seite beschränke ich mich daher auf IBM und SUN Microsystems und deren Sicht auf Service Orientierte Architekturen. Beide Unternehmen vertreiben eine ganze Produktpalette die alle Teile des Software Lifecycles abdeckt. Ich werde im folgenden nur zwei Tools für Modellierung und Implementierung genauer vorstellen; Den Rational Software Architect von IBM basierend auf Eclipse und Netbeans mit dem Enterprise Pack 5.5 welches die IDE um Funktionen aus dem Java Studio Enterprise 8 erweitert. Als Infrastruktur kommt beidemale ein J2EE Application Server zum Einsatz: Die Websphere Platform von IBM und von SUN Microsystems der SUN Java System Application Ser-

ver.

In den Dokumenten und Papers wird gerne darauf hingewiesen dass Webservices nur eine möglich Realisierung einer Service Orientierten Architektur sind. Als Alternativen sind CORBA, DCOM, JAVA RMI mit JINI uva. denkbar aber ohne die Vorzüge von öffentlichen Standards wie SOAP, WSDL usw. Aufgrund der Einschränkungen die ich für dieses Dokument bis hierher gemacht habe, werde ich die Begriffe Service und Webservice synonym verwenden.

2 Related Work

Wie in der Einführung schon angeschnitten reicht der SOA Dunstkreis von der Modellierung der Geschäftsprozesse über die Implementierung der Services bis zu Deployment und Monitoring auf einer bestimmten Infrastruktur. Jeder Anbieter von SOA Lösungen hat zu seinen Produkten maßgeschneiderte Whitepapers und Produktdokumentationen. An Literatur mangelt es somit nicht. Für diese Arbeit waren zwei Dokumente ausschlaggebend: *IBMs SOA Foundation - An Architectural Introduction and Overview* [1] und das Äquivalent von SUN *Service-Oriented Architecture and Webservices: Concepts, Technologies and Tools* [12].

3 IBMs SOA Foundation

The Primary goal of Service Oriented Architecture (SOA) is to align the business world with the world of information technology (IT) in a way that makes both more effective

IBM SOA Foundation, Chapter 1, Page 6 [1]

Grundlage für jede SOA ist ein Business Design. Als Business Design versteht man eine Beschreibung wie das Unternehmen funktioniert, die Geschäftsprozesse, die Organisation der Mitarbeiter und der Finanzen sowie die kurzfristigen und langfristigen Unternehmensziele.¹ Das Business Design dient als Kommunikationsgrundlage für Mitarbeiter und Entwickler um zu verstehen was und wie

¹Frei nach [1] Seite 6, Absatz 2

etwas im Unternehmen gemacht wird. Im Lauf des SOA Entwicklungszyklus kommt dem Business Design essentielle Bedeutung zu; Das Design des Informationssystems wird aus dem Business Design automatisiert abgeleitet. So können Änderungen im Business Design leichter in die IT übernommen werden und aus dem laufenden Informationssystem sind Rückschlüsse auf den Unternehmenszustand möglich. Mehrnoch als passives Monitoring sollen Optimierungen soweit wie möglich automatisiert in das Business Design übernommen werden.

Es ist klar dass man soviel Automatisierung nicht geschenkt bekommt; Während des ganzen Lebenszyklus, von der Entwicklung über Wartung im laufenden Betrieb bis zur Anpassen an neue Aufgaben, müssen eine ganze Palette von Tools miteinander kommunizieren. Es gibt auch heute gute Tools für Teilaufgaben im Lifecycle, wie die Modellierung von Geschäftsprozessen oder die Implementierung von Geschäftslogik in Java. Schwachstelle aller etablierten Tools ist die mangelnde Kommunikation untereinander. Soweit die Vision von IBM zum Thema SOA. Zur Realisierung müssen eine Reihe von Voraussetzungen erfüllt sein:

- Das Business Design muss in einer formalisierten Sprache verfasst werden. Vorzugsweise eine XML Anwendung. Aus bunten Visio Diagrammen lässt sich schwer ein Informationssystem ableiten.
- Es muss eine Methodik geben um aus einem Business Design Artefakte eines Informationssystem zu generieren. Zum Beispiel wie aus Aktivitäten in einem Prozess Webservices und Interfaces generiert werden. Wie einzelne Webservices zu einem Prozess orchestriert werden, usw.
- Es muss eine Infrastruktur geben auf der die Artefakte (Webservices im speziellen) laufen und miteinander kommunizieren können. Allgemein als Enterprise Service Bus bezeichnet.
- Es muss eine Möglichkeit geben das Business Design mit dem Informationssystem abzugleichen und Änderungen durchzuführen.
- Es muss möglich sein das System im laufenden Betrieb zu beobachten und mit den Geschäftszielen zu vergleichen. zB Anzahl der abgeschlossenen Aufträge.

3.1 SOA Lifecycle

Bevor man die Tools evaluiert muss man den Lifecycle kennen für den sie entwickelt wurden. IBM hat für die Entwicklung, Wartung und Monitoring von SOA Anwendungen einen eigenen SOA Lifecycle aufgestellt.

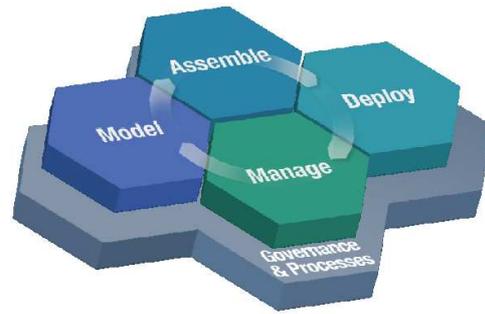


Abbildung 1. IBM SOA Lifecycle, Quelle: [1]

Der SOA Lifecycle startet bei der Modellierung eines Business Designs. Aus dem Business Design wird ein Modell für das zu erstellende Informationssystem abgeleitet. Der SOA Lifecycle ist iterativ, somit terminiert die Modellierungsphase nicht mit einem entgeltigen Design. In der nächsten Iteration werden die Daten aus der *Manage* Phase zur Verfeinerung des Business Designs herangezogen.

In der *Model* Phase wurde das Design des Informationssystem erstellt oder verfeinert und Artefakte generiert. In der *Assemble* Phase findet der Grossteil der Entwicklungsarbeit statt. Die Artefakte müssen nun zusammengesetzt werden. Dazu gehört auch bestehende Subsysteme einzubinden. Einige Systeme können vielleicht einfach eingebunden werden, manche Funktionen müssen komplett neu geschrieben werden.

Der nächste Schritt ist das Aufsetzen und Vorbereiten einer Infrastruktur auf der die SOA Anwendung laufen soll. Der zweite Teil der *Deploy* Phase ist das Hosten der Anwendung und der Services selbst.

Die Letzte Phase in einer Iteration ist die *Manage* Phase. Ist erstmal eine passende Infrastruktur installiert und die Anwendung darauf gehostet beginnt Wartung und Monitoring. Dazu gehören Überwachung der Performance, Antwortzeiten einzelner Services und Auswerten von Logs. Neben den reinen technischen Aspekten ist auch der Abgleich mit dem Business Model Teil der *Manage* Phase. Dazu zählen Wartungsarbeiten damit die Anwendung wieder den Vorgaben aus dem Business Modell entspricht.

3.2 SOA Infrastructure and Tools

Grundlage für eine SOA ist ein umfangreiche Middleware; Der Enterprise Service Bus. IBM wird nicht müde zu erwähnen dass SOA technologie-neutral zu sehen ist, dennoch bietet sich die J2EE Platform als mögliche ESB Grundlage geradz zu an. So ist auch die IBM Lösung, die

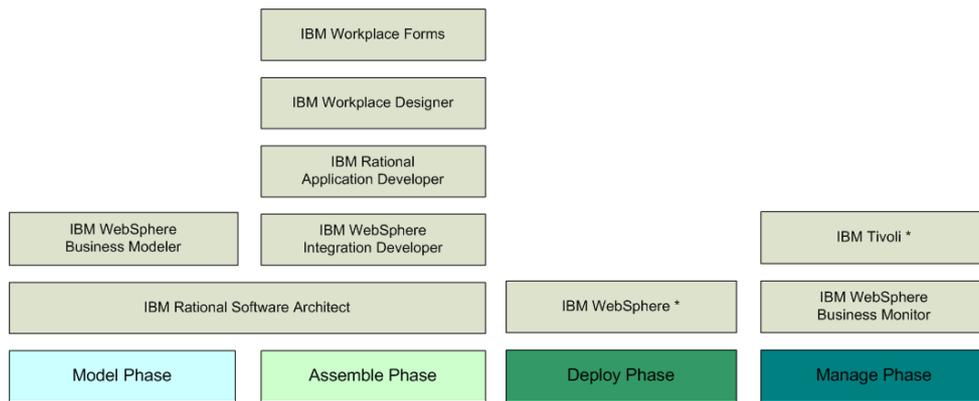


Abbildung 2. IBM Tools für die Phasen des SOA Lifecycle

WebSphere Platform, ein J2EE Application Server mit zusätzlichen Komponenten für Deployment, Wartung, Monitoring, usw. Zu jeder Phase im SOA Lifecycle bietet IBM auch die passenden Tools an (siehe Fig. 2). Jedes Tool legt den Schwerpunkt auf eine Phase im Lifecycle, überschneidet sich aber idR. mit Tools aus vorhergehenden oder nachfolgenden Phase.

3.3 IBM Rational Software Architect

Der IBM Rational Software Architect (RSA) ist im SOA Lifecycle in der *Model* und *Assemble* Phase angesiedelt. In der *Model* Phase deckt der RSA die technischen Aspekte der SOA Modellierung ab. Für die Ausarbeitung des Business Design gibt es den IBM WebSphere Business Modeller [4].

Der RSA basiert auf Eclipse und kann auch wie Eclipse für die Entwicklung von Standard Java Anwendungen verwendet werden. Zusätzlich unterstützt der RSA die Entwicklung von J2EE Anwendungen und Webprojekten mit Java Server Pages und Servlets oder auch statischem HTML. Bei der Default RSA Installation wird auch die IBM WebSphere Middleware mitinstalliert und für die RSA Entwicklung konfiguriert. Alternative Middleware lässt sich auch per Hand nachkonfigurieren. Neben Java unterstützt der RSA auch die C und C++ Entwicklung. Grundlage dafür sind die C/C++ Development Tools (CDT) Plugins [3] in Version 2.0.2 wie sie auch in der freien Eclipse Version Verwendung finden. Damit ist der RSA nicht mehr ganz auf dem neuesten Stand, das aktuelle Release zu diesem Zeitpunkt ist 3.0.1 (Nov. 2005) oder für die 2er Linie Version 2.1 (Dez. 2004). Als C/C++ Compiler kommt unter Linux üblicherweise GCC zum Einsatz, alternative Compiler lassen sich für die CDT konfigurieren.

Wie Eclipse kann auch der RSA mit Plugins erwei-

tert werden. IBM pflegt dazu eine eigene Webseite [5]. Die gelisteten Plugins decken alle Kategorien des Entwicklungszyklus ab; Modellierung, Entwicklung und Deployment, UI Tools, Qualitätssicherung uvm. Die *Ready for IBM Rational Software* Validierung garantiert das reibungslose Zusammenspiel von IBM Rational Software und Plugins der IBM Business Partner. Weil der RSA auf Eclipse basiert lässt er sich auch mit Plugins erweitern die für die frei Eclipse IDE geschrieben wurden [6] (siehe Fig. 3). Wie auch bei der freien Eclipse Version üblich, muss die Kompatibilität zwischen Plugin und den verschiedenen Versionen geachtet werden. Der RSA Trial in Version 6.0 basiert auf Eclipse 3.0.1. Die Versionsnummern aller anderen Komponenten sind über das Hilfe Menü, Eintrag Info über Rational Software Development Platform Trial abrufbar. Entsprechende Hinweise zur Kompatibilität finden sich meistens auf der Homepage des Plugin Herstellers. Die Plugin Installation gestaltet sich bei Eclipse und RSA gleich; Das gepackte Plugin entpacken und die Ordner oder JAR Dateien in die Feature bzw Plugin Ordner der IDE Installation kopieren.²

Für die SOA Entwicklung finden sich im IBM Plugin Central auch zwei interessante Plugins, beide von exaltec Software [7]; Der b+ J2EE SOA Application Generator und der b+ SOA J2EE Competitive Migrator. Beide Plugins sind kommerziell und müssen gegen Lizenzgebühr erstanden werden. Der Application Generator verspricht die Erzeugung einer SOA ohne dass die Entwickler eine Zeile Code schreiben müssten.³ Der Competitive Migrator unterstützt das Reengineering von

²Wenn nicht anders spezifiziert befinden sich die Feature und Plugin Ordner einer Linux Installation im Verzeichnis `/opt/IBM/Rational/SATrial/6.0/eclipse`

³ "With b+, business analysts and developers can create entire applications without necessarily having to write a single line of code."
Quelle: <http://www.exaltec.com/appgenerator.html>

Legacy Anwendungen und Transformation von Client-Server Architekturen in SOAs.

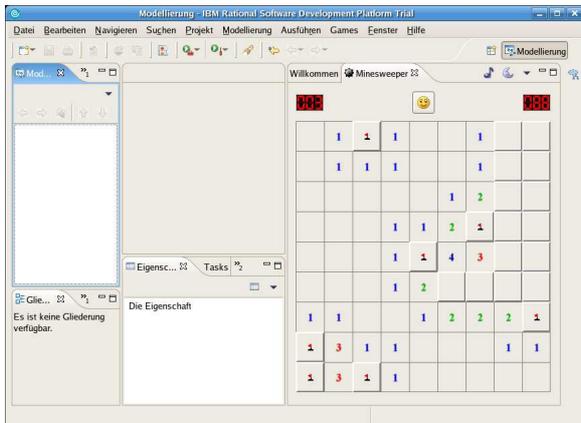


Abbildung 3. Minesweeper Plugin für Eclipse funktioniert auch im RSA :)

Entsprechend der Positionierung im SOA Lifecycle ist die Hauptaufgabe des RSA die Modellierung von Architektur und Anwendung im Detail. Modellierungssprache ist UML 2.0. Dem Modellierungswerkzeug sind im RSA keine Grenzen gesetzt, modelliert werden kann alles von C++ Klassen, POJOs bis J2EE Module. Der RSA unterstützt die Verwendung von vordefinierten Design Patterns und die Erstellung von eigenen Pattern. Weiters kann aus dem UML Modell in wenigen Schritten Java, EJB oder C++ Code generiert werden.



Abbildung 4. RSA: Aus UML Diagramm Code generieren

4 SUNs Perspective on SOA

Die Umsetzung des SOA Paradigmas in der Welt von SUN verhält sich ähnlich wie bei IBM. Grundlage für eine

SOA ist ein Enterprise Service Bus. Ein mögliche ESB Referenzimplementierung ist der OpenESB [8]. Kern des OpenESB ist die Java Business Integration Technology (JBI), womit die Integration und Kommunikation von SOA-like Services erst möglich wird. JBI selbst ist im JSR 208 [9] spezifiziert. In diesem Paper wird JBI nicht weiter besprochen, der interessierte Leser möge die SUN Nettalk Reihe zum Thema JBI besuchen [10]. Host für den OpenESB ist entweder der SUN Java Systems Application Server oder die Open Source Variante von Project Glassfish [11].

Die Aussöhnung mit dem einstigen Erzfeind aus Redmond 2004, hat für die SOA Entwicklung Früchte getragen. Ergebnis ist die Web Service Interoperability Technology (WSIT). WSIT ist eine Open Source Implementierung um die Interoperabilität zwischen Java EE und .NET zu verbessern und gemischte SOA Anwendung zu ermöglichen. Für die SOA Entwicklung mit Netbeans 5.5 Beta steht ein WSIT Plugin zum Download bereit. Damit lassen sich .NET kompatible Web Services und Web Service Clients entwickeln.

4.1 SUN Java IDEs and Frameworks Overview

Die Netbeans [13] IDE ist eine quelloffene Entwicklungsumgebung für Java und lässt sich mit Plugins für andere Sprachen erweitern. Sie bildet den Kern der Entwicklungsumgebungen von SUN Microsystems für die Java Plattform.

- Netbeans stammt ursprünglich von SUN und wird auch noch von SUN gepusht. Die Netbeans IDE gibt es optional mit dem J2SE SDK oder J2EE inklusive SUN Java Systems Java Application Server im Bundle zum Download. Die Netbeans IDE unterstützt die Entwicklung von Multitier J2EE Anwendungen und Webservices auf J2EE Basis.
- Das Netbeans Enterprise Pack 5.5 [14] erweitert die Netbeans IDE um Funktionalität aus der Java Studio Enterprise IDE. Neu für Netbeans ist ein UML Modellierungswerkzeug welches optional Java Code erzeugt und Reverse Engineering von Java Anwendungen in ein UML Modell ermöglicht. Für die SOA Entwicklung interessant ist der neue BPEL Designer für die Service Orchestrierung.
- Der Java Studio Creator 2 [15] richtet sich an Entwickler von Webanwendung auf Basis von Java Server Pages und dem Java Server Faces Framework. Der Creator ist eine einfach zu bedienende Java Alternative zu Microsofts Visual Webdeveloper für ASP.NET.

Für die SOA Entwicklung ist der Creator nur soweit interessant dass sich mit wenigen Handgriffen [17] existierende Webservices nutzen lassen. Der Java Studio Creator ist für registrierte User kostenlos, aber nicht Open Source.

- Das Java Studio Enterprise 8 ist die schwergewichtigste Java IDE im Portfolio von SUN. Wie der Name schon erahnen lässt liegt der Einsatzzweck der IDE im J2EE Bereich. Das Studio basiert zwar auf Netbeans aber es fehlen beispielsweise der Java Swing GUI Editor Matisse welcher seit Version 5 Teil der Netbeans IDE ist. Mit dem Studio lässt sich praktisch alles rund um J2EE realisieren. In Version 8 fehlen dem Studio noch der BPEL Designer welcher Teil des Enterprise Packs für Netbeans ist.
- Das Web Services Developer Pack [16] ist ein Bundle von Tools und Frameworks rund um die Entwicklung von XML Anwendungen, Web Services und Webanwendungen⁴. Wichtiger Bestandteil des WSDP ist der "integrated Stack" für Webservices, bestehend aus dem Java API for XML Web Services (JAX-WS), der Java Architecture for XML Binding (JAXB) und dem SOAP with Attachments API for Java (SAAJ). Host Platform kann auch hier wieder der SUN Java System Application Server, die Open Source Variante vom Project Glassfish oder nur ein Tomcat for WSDP 5.0 Server als Container für Servlets und JSPs. Für letzteren bietet SUN keinen Support an.

- UML Werkzeug
- XML Schema Tools
- SOA Module, BPEL Engine für JBI
- SUN Java System Application Server 9
- OpenESB Starter Kit
- SUN Java System Access Manager 7.1
- SUN Java System Access Manager Policy Agent 2.2

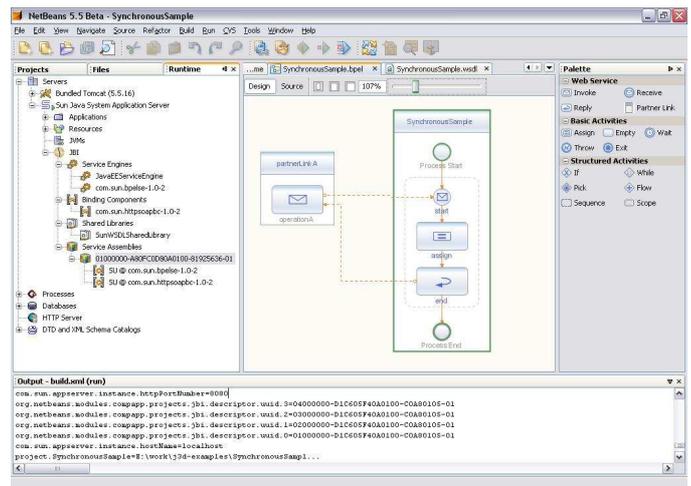


Abbildung 5. Netbeans: BEPL Editor - Quelle: Netbeans Homepage

4.2 Netbeans 5.5 Beta with Enterprise Pack 5.5 Early Access

Die Geschichte von Netbeans geht zurück bis 1996, als tschechische Studenten eine Java IDE schrieben. Die kleine Firma Netbeans wurde von SUN übernommen und die IDE unter dem Name *Forte* vermarktet. Im Juni 2000 wurde aus *Forte* die Quelloffenen *Netbeans* IDE. Seit damals hat Netbeans grossen Zuspruch gefunden und wird von der Community und SUN weiterentwickelt. Mit der Zeit bekam Netbeans auch J2EE Support und mit Matisse einen der besten⁵ Java GUI Editoren. Ähnlich wie Eclipse ist auch Netbeans über Plugins erweiterbar.

Das Enterprise Pack von SUN ist der neueste Zuwachs für Netbeans und erweitert die Open Source IDE um Funktionen aus dem Enterprise Studio. Voraussetzung für die Installation des Enterprise Packs ist eine aktuelle Netbeans 5.5 Beta Version. Das Enterprise Pack besteht aus folgenden Komponenten:

Zusammenfassen kann man sagen dass Netbeans mit dem Enterprise Pack 5.5 alles bietet um eine SOA zu planen, implementieren, testen und zu deployen. Im direkten Vergleich mit dem RSA von IBM ist Netbeans + EP mindestens gleichwertig. Auch der Netbeans UML Editor muss sich vor IBMs RSA nicht verstecken. Netbeans mit EP beherrscht wie der RSA die Code Generierung aus UML heraus. Zusätzlich unterstützt das Enterprise Pack UML Werkzeug das Reverse Engineering von existierenden Java Anwendungen in ein UML Modell. Ähnlich wie im RSA gestaltet sich die Anwendung der etablierten Design Patterns⁶ und J2EE Pattern im speziellen.

4.3 The Future: SOA with Open Source Components from SUN

Open Source ist die Zukunft; SUN hat die Zeichen der Zeit erkannt und engagiert sich massgeblich im Open Source Bereich. Auf der JavaONE Konferenz hat SUNs neuer

⁴Frei nach [16] 1. Absatz

⁵Persönliche Meinung des Autors

⁶Gang of 4

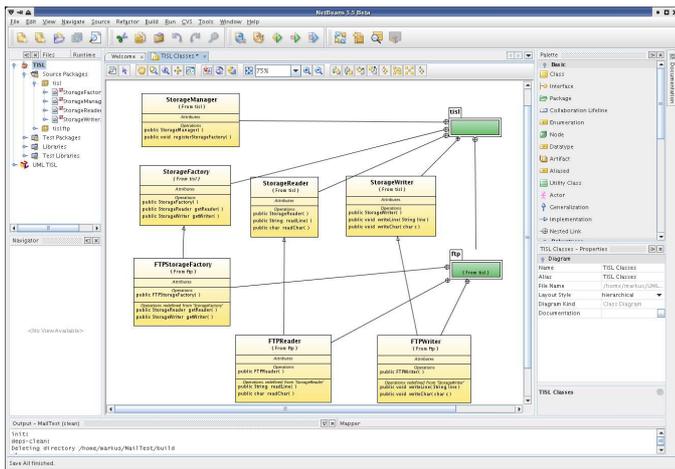


Abbildung 6. Netbeans: UML Editor erzeugt parallel Java Code

CEO, Jonathan Schwarz, auch deutlich gemacht dass SUN bestrebt ist eine komplette Open Source SOA Lösung anzubieten. Viele der notwendigen Komponenten sind bereits Open Source:

- Das WSIT Pack fördert .NET Kompatibilität und wurde unter CDDL [18] veröffentlicht
- Der OpenESB ist eine Referenzimplementierung eines Enterprise Service Bus, ebenfalls Open Source unter CDDL.
- Das Glassfish Projekt bietet einen vollwertigen J2EE Application Server. Auch hier gilt wieder die CDDL.
- Netbeans wurde schon im Juni 2000 Open Source, damals noch unter SUN Public License [19].
- OpenSolaris [20] ist die Open Source Version von Solaris 10, der System V Unix Variante SUN. Die Veröffentlichung des Source Codes hatte sich wegen rechtlicher Fragen etwas länger hingezogen. Schlussendlich wurde aber auch Solaris unter CDDL Open Source.
- Sogar die Hardware [21] ist teilweise Open Source. SUN hat das Design des UltraSparc T1 im Verilog Source Code unter GNU General Public License veröffentlicht.

Nur die Gretchenfrage konnte oder wollte Schwarz auf der JavaONE noch nicht beantworten; "Wann wird Java selbst Open Source ? " Aber er liess mit der Aussage aufhorchen dass keine Frage mehr ist *Ob* Java Open Source

wird, nur mehr *Wie*⁷. Java ist für Forschung und Entwicklung schon längere Zeit quelloffen, aber weder die SUN Community Source License (CSCL) noch die Java Research License (JRL) gewähren die notwendigen Rechte um als vollwertige Open Source Lizenz anerkannt zu werden. Es bleibt noch abzuwarten ob die Vision einer puren Open Source SOA Lösung Realität wird; Die Zeichen stehen gut.

5 Conclusion

SOA ist da und wird so schnell nicht mehr verschwinden. Grundlage für SOA sind *offene* Standards, an denen die führenden Softwareproduzenten mitgearbeitet haben und allgemein akzeptiert sind. An SOA wird in den nächsten Jahren kein Weg mehr vorbei führen. Trotz der vielbeschworenen Technologie-Unabhängigkeit und Interoperabilität untereinander werden sich Architekten und Entwickler die Frage stellen müssen auf welche Middleware sie setzen. Die Frage der Tools beantwortet sich (momentan) durch die Wahl der Middleware. In diesem Paper wurde 2 Ansätze vorgestellt; Die IBM Version mit Websphere Middleware und den IBM Rational Produkten und die Open Source Variante mit Software aus dem SUN Dunstkreis.

	Netbeans 5.5 with Enterprise Pack	IBM Rational Software Architect
Basierend auf	Netbeans	Eclipse
Erweiterbar	Ja, Plugins für Netbeans	Ja, Plugins für Eclipse
Java Support	J2ME, J2SE, J2EE	J2SE, J2EE
UML	UML 2.0, grafische Editor	UML 2.0, grafischer Editor
UML Patterns	Gang of 4, EJB, eigene Patterns	Gang of 4, EJB, eigene Patterns
Code aus UML generieren	Für Java	Für Java und C++ Klassen
BPEL Editor	Ja	Nein
Grafischer XML Schema Editor	Ja	Nein
Bundle mit J2EE Server	SUN Java System Application Server 9	Websphere
OpenSource	Ja, Sun Public License	Nein
Verfügbar für	Windows, Linux, Solaris, MacOS X	Windows, Linux

Quellen:
 IBM Rational Software Architect Specification,
<http://www3.software.ibm.com/ibmdl/pub/software/rational/web/datasheets/rsa.pdf> Online 4.6.2006
 Netbeans Enterprise Pack Features,
<http://www.netbeans.org/products/enterprise/features.html> , Online 4.6.2006

Abbildung 7. Vergleich Netbeans Enterprise Pack 5.5 Beta mit IBM Rational Software Architect

Beide SOA Varianten haben ihre Vorzüge und Nachteile. Die IBM Tools und Middleware schleppt noch viele Rational Altlasten mit. Obwohl IBM für SOA einen eigenen Lifecycle aufgestellt hat, schmeckt man noch aus allen Produkten den schwergewichtigen RUP heraus. Ein eigener SOA Unified Process würde Sinn machen.

⁷"It's not a question of whether we'll open source Java, now the question is how." - Jonathan Schwarz, CEO of SUN

Eine reine Open Source SOA Lösung gibt es noch nicht weil wichtige Teile, wie Java, noch nicht Open Source sind. Selbst wenn man über diesen Mangel hinwegsieht steht die Open Source Lösung noch auf wackeligen Beinen. Die zahlreichen Beta, Preview und Early Access Modifier in der Liste der Open Source Komponenten lassen von der Verwendung im Produktivsystem Abstand nehmen. In ein oder zwei Jahren jedoch wird die Open Source Variante eine ernsthafte Alternative zu den kommerziellen SOA Lösungen sein.

Literatur

- [1] Rob High Jr., Stephen Kinder, Steve Graham, *IBM's SOA Foundation, An Architectural Introduction and Overview*, Version 1.0, November 2005, Online 16.4.2006
<http://www-128.ibm.com/developerworks/webservices/library/ws-soa-whitepaper/>
- [2] IBM Service Oriented Architecture (SOA): Products, Online 20.5.2006
<http://www-306.ibm.com/software/solutions/soa/offerings.html>
- [3] C/C++ Development Tools, Online 21.5.2006
<http://www.eclipse.org/cdt/>
- [4] IBM Websphere Business Modeller, Online 20.5.2006
<http://www-306.ibm.com/software/integration/wbimodeler/index.html>
- [5] developerWorks: Rational Software Downloads: plugins, Online 21.5.2006
<http://www-128.ibm.com/developerworks/websphere/downloads/plugin/>
- [6] Eclipse Plugin Central, Online 21.5.2006
<http://www.eclipseplugincentral.com/>
- [7] Exaltec Software, Online 21.5.2006
<http://www.exaltec.com/>
- [8] OpenESB, Online 20.5.2006
<http://open-esb.dev.java.net/public/plans.html>
- [9] JSR 2008: Java Business Integration (JBI), Online 20.5.2006
<http://www.jcp.org/en/jsr/detail?id=2008>
- [10] SUN Net Talks On Demand: JBI Series
<https://communications.sun.com/dialog/singleeventpage.do?eventId=6320>
<https://communications.sun.com/dialog/singleeventpage.do?eventId=6321>
<https://communications.sun.com/dialog/singleeventpage.do?eventId=6322>
- [11] Project Glassfish, Online 21.5.2006
<https://glassfish.dev.java.net>
- [12] Ed Ort, *Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools*, April 2005, Online 16.4.2006
<http://java.sun.com/developer/technicalArticles/WebServices/soa2/>
- [13] Netbeans
<http://www.netbeans.org>
- [14] Netbeans Enterprise Pack 5.5
<http://www.netbeans.org/products/enterprise/index.html>
- [15] Java Studio Creator 2
<http://developers.sun.com/prodtech/javatools/jscreator/>
- [16] Web Service Developer Pack 2, Online 21.5.2006
<http://java.sun.com/webservices/jwsdp/index.jsp>
- [17] Access Web Services with Java Studio Creator 2
<http://developers.sun.com/prodtech/javatools/jscreator/learning/tutorials/2/webservices.html>
- [18] Common Development and Distribution License
<http://www.opensource.org/licenses/cddl1.php>
- [19] SUN Public License <http://www.netbeans.org/about/legal/spl.html>
- [20] Open Solaris <http://www.opensolaris.org>
- [21] Open Sparc <http://opensparc.sunsource.net/nonav/index.html>

Copyright (c) 2006 Markus Nöbauer

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled GNU Free Documentation License.