



JavaTMmagazin

Internet & Enterprise Technology

javamagazin.de



**Auf
CD**

Alle Infos auf S.3

Open Source Java! Jetzt auf CD!

- Java Compiler
- Java HotSpot Virtual Machine
- Java jrtreg
- ME Framework
- PhoneME



Mit den W-JAX-Sessions (Audio)

- Enterprise developers, basement hackers, and the Java Platform (Tim Bray)
- Chancen und Risiken von Ajax in Fachanwendungen (Markus Eisele)

Java 6

- Wie gerechtfertigt ist der Versionssprung?
- Neue Features im Fokus: Web Services, JDBC 4 & Java DB

Open Source Java

**Was ändert sich für den
Java-Entwickler?**

JSF: Performance à la MyFaces

Spring & Groovy: Enterprise Java meets Scripting

Neu Architect Corner: Warum über Architekturen nachdenken?

Enterprise **ARCHITEKTUR**
MAGAZIN

- Transformation von Kernbankensystemen – oder:
Wie überwindet man Herausforderungen von Großprojekten
- Abnahmen: Verhinderer des Projekterfolges?

Datenträger
enthält nur
Lehr- oder
Infopro-
gramme

D 45867



W-JAX 06 und Enterprise Architektur Konferenz vom 6. bis 10. November 2006 in München

Technologische Impulse

■ VON ARNO HAASE

In der Woche vom 6. bis 10. November fand im ArabellaSheraton Grand Hotel in München die fünfte W-JAX statt. Der folgende Konferenzbericht gibt eine subjektive Sicht auf die Konferenz wieder. Er hat keinen Anspruch auf Vollständigkeit, beleuchtet dafür aber die Veranstaltungen aus verschiedenen Richtungen.

Bereits zum zweiten Mal lief parallel zur Konferenz für Java, Enterprise-Architekturen und SOA die Enterprise Architektur Konferenz (EAKon), die den gesamten Software-Lifecycle adressierte und Softwarearchitekturen, Konzepte, serviceorientierte Architekturen (SOA), agile Vorgehensweisen sowie Team- und Projektmanagement fokussierte. Die W-JAX hingegen, als „nicht mehr ganz so kleine“ Schwesterkonferenz der JAX, thematisierte wie in den Jahren zuvor Enterprise-Java-Technologien, Frameworks und Entwicklungstechniken. Dieses Jahr ging die Veranstaltung zum ersten Mal über fünf Konferenztage.

Mit 750 Teilnehmern war die Konferenz sehr gut besucht, ein Zuwachs um 25 Prozent gegenüber dem Vorjahr. Auch die Ausstellung war dieses Mal mit 30 Ausstellern deutlich größer geworden. Ich hatte befürchtet, dass der persönliche Charakter der Konferenz unter dem Zuwachs leiden könnte, und tatsächlich war es deutlich voller und gedrängter. Trotzdem gab es genug Raum, um einfach mal zu plaudern und Leute kennen zu lernen, die Konferenz blieb trotz der Größe persönlich.

Wie schon gesagt, ging die Konferenz dieses Jahr zum ersten Mal über fünf Tage, und dementsprechend stand den Teilnehmern eine größere Auswahl an Sessions zur Verfügung. Am ersten und letzten Tag der Konferenz gab es in bewährter Manier ganztägige Power Workshops. Parallel dazu fanden der Spring Day, der SOA Management Day und der BPM Day

statt. Das Ganze war wieder gespickt mit Keynotes, in denen Speaker von international tätigen Großunternehmen ihre Visionen darlegten.

Ballroom

Am Dienstagabend gab es erstmals den W-JAX Ballroom, ein offenes Forum zum Austausch. Es war ein Experiment, im großen Saal des Konferenzhotels – er trägt den Namen „Ballsaal“, daher der Name der Veranstaltung – runde Tische aufzustellen, auf denen jeweils ein Schild mit einem Themenschwerpunkt stand.

Die Idee dahinter war, dass jeder, der sich für ein Thema interessierte – z.B. Hibernate, modellgetriebene Softwareentwicklung, Spring, aber auch „weiche“ Themen wie agile Softwareentwicklung – zum entsprechenden Tisch gehen konnte, um dort mit Gleichgesinnten zu diskutieren. Die Veranstaltung war gut besucht, und es gab viele interessante Gespräche und Diskussionen. Eine ganze Reihe dieser Gespräche zog sich bis spät in den Abend hinein. Dieses neue Konzept ist sicherlich wiederholungswürdig.

Im Rahmen des Ballroom wurde auch der fünfte Geburtstag von Eclipse gefeiert: Fünf Jahre zuvor hatte IBM eine Reihe von Journalisten eingeladen, um das bis dahin große Geheimnis um das neu entwickelte System zu lüften. Die damalige Pressekonferenz war am Dienstag zwar eigentlich schon fünf Jahre und zwei Tage alt, aber der 7. November war aus irgendwelchen Gründen weltweit zum of-

fiziellen Geburtstag von Eclipse erkoren worden.

Jedenfalls rollte zu Beginn des Ballrooms ein Speisewagen mit großer Geburtstagstorte, Tischfeuerwerk und Wunderkerzen in den großen Raum, aus den Lautsprechern ertönte feierliche Musik – ein klarer Akzent und eine angemessene Art, eines der für Java bedeutsamsten Open-Source-Projekte zu würdigen.

Open-Source-Lizenzmodell für Java

Die Eröffnungs-Keynote der Veranstaltung hielt Tim Bray, Director of Web Technologies bei Sun Microsystems und verantwortlich für die Einbindung von



Abb. 1: Sebastian Meyer gibt die Richtung vor.

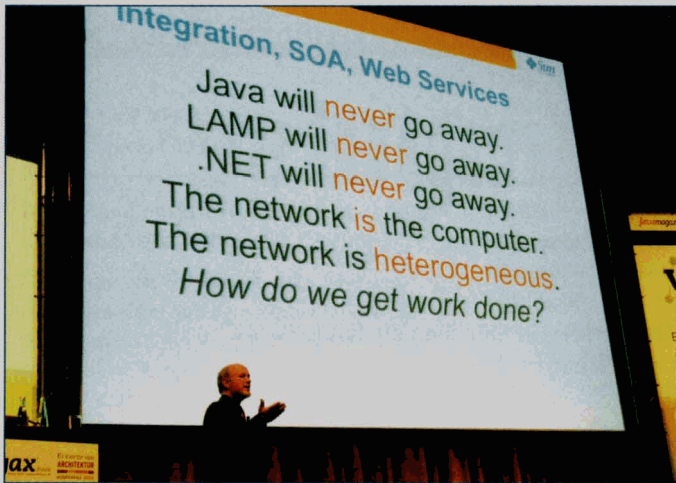


Abb. 2: Tim Bray prophezeit „Java forever“.



Abb. 3: Voller Ballsaal während der Keynote von Tim Bray

Script-Sprachen in Java. Er kündigte an, dass ursprünglich ebendieser Dienstag das Datum für Sun gewesen wäre anzukündigen, unter welchem Open-Source-Lizenz-Modell Java zukünftig angeboten werde. Allerdings sei die Wahl in den Vereinigten Staaten dazwischen gekommen. Er gab bekannt, dass Sun am Montag nach der Konferenz die Einzelheiten bekanntgeben würde, was ja inzwischen geschehen ist: Java wird unter die GPL gestellt, zunächst Compiler, Hotspot und die Standardbibliothek, der Rest der Plattform soll 2007 folgen. Im Zuge von Open Source Java sei es wahrscheinlich, so Bray, dass es „Forks“ der Java-Plattform geben werde. Das müsse nicht unbedingt schlecht sein und wäre z.B. für Sun ein Anreiz zu intensiverer Arbeit. Bray erwartet auch, dass Open Source Java eine gewaltige Auswirkung auf Open-Source-Desktop-Systeme haben werde, die bis dato auf C++ basierten.

Von Ruby und PHP lernen

Im Zentrum seiner Keynote stand dann aber abzuwägen, wie es in Zukunft weitergehen werde mit Java. Zweifellos sei für ihn Java unübertroffen in Bereichen wie Performance, Concurrency, Deployment, Dokumentation, Security, Bibliothekenvielfalt und Community-Größe und die Java Enterprise Edition (Java EE) mit ihrer „Ease of Development“-Ausrichtung auf einem guten Weg, das sei aber nicht genug. Deswegen sei es wichtig, Sprachen wie PHP oder Ruby genau zu studieren und von ihnen zu lernen. Bray sieht das Dreigestirn, bestehend aus JVM, Java-Sprache und Ja-

va-APIs in Zukunft aufgeteilt in JVM, Java APIs und einer Vielfalt von Programmiersprachen wie Java, Ruby, JavaScript, Python, PHP etc. So ist für die nahe Zukunft mit einer Ruby on Rails-Implementierung für Java zu rechnen.

Bewährte Technologien

Ein Trend, der sich auch dieses Jahr fortsetzt, ist das fortlaufende Interesse an soliden, bewährten technologischen Plattformen. Das war auf der W-JAX ganz besonders Spring, welches mit dem separaten Spring Day prominent vertreten und auch sonst Gegenstand einer Vielzahl von Sessions war. Dabei ging es zum einen um die technische Seite: Wie funktioniert AOP mit Spring, wie sieht Spring intern aus, was sind die neuen Features etc.? Daneben sind aber inzwischen Architektur- und Vorgehensthemen getreten: Wie verwende ich die Features von Spring im Kontext einer realen Anwendung und was ist dabei zu beachten?

Daneben stand Hibernate als De-facto-Standard für Persistenz mit Java – auch auf Basis von JPA – und auch hier ist neben der Vorstellung der technischen Aspekte zunehmend Interesse an einer guten Verwendung im Kontext größerer Systeme getreten.

Ein drittes großes Thema war auch wieder die Eclipse-Plattform. Das bedeutet zunächst die Rich Client Plattform (RCP), die nach wie vor eine der attraktivsten Plattformen für Rich-Client-Anwendungen ist. Daneben ist aber – sowohl bei der Gewichtung der Sessions als auch

im Interesse der Teilnehmer – das Thema „Server Side Eclipse“ getreten.

Dabei geht es darum, OSGi – die von Eclipse bekannten „Plug-ins“ – als Komponentenmodell für serverseitige Anwendungen und dabei Equinox als die Eclipse-Implementierung des OSGi-Standards zu verwenden. Das klingt im ersten Augenblick vielleicht überraschend, weil man Eclipse und Plug-ins gewohnheitsmäßig mit clientseitiger Entwicklung verbindet. Das OSGi-Komponentenmodell bietet mit sehr loser Kopplung und dynamischer Bindung, individuellem Lebenszyklus für jede Komponente und nicht zuletzt dem Extension-Point-Mechanismus für erweiterbare Schnittstellen eine ganze Reihe von Features, die auch für serverseitige Entwicklung interessant sind. Es lohnt sich, das Thema im Auge zu behalten.

OSGi und Spring sind dabei keineswegs Konkurrenten, die sich gegenseitig ausschließen: Man kann sie miteinander verbinden, z.B. indem man feingranulare Spring-Komponenten in OSGi-Bundles einbettet. Umgekehrt gibt es Implementierungsansätze, OSGi-Bundles in Spring-Komponentenkonfigurationen einzubetten.

Ein vierter großer technischer Trend sind Webanwendungen. Was auf den ersten Blick wie ein alter Hut aussieht, ist wieder – oder immer noch – ein heißes Thema. Webanwendungen erobern immer mehr den Desktop, und das bedeutet den Einsatz von Ajax, um den gewohnten Bedienkomfort sicherstellen zu können. Dabei gibt es inzwischen eine Reihe von

Bibliotheken und Frameworks – kommerzielle wie freie – aus denen man das Passende auswählen kann. Die Bandbreite reicht dabei von einfachen JavaScript-Bibliotheken bis hin zu kompletten Anwendungs-Frameworks.

Interessant an diesen Technologien ist, dass keine von ihnen auf ein Standardisierungsgremium zurückgeht: Sie haben durchweg als Open-Source-Implementierung ihr Leben begonnen. Sie erfreuten sich dann zunehmender Beliebtheit in Projekten und sind (mit Ausnahme von Ajax, wo sich bisher kein Standard abzeichnet) so zum De-facto-Industriestandard avanciert. Im Falle von Hibernate hat das sogar dazu geführt, dass sich JPA als formaler Standard weitgehend an der gewachsenen Open-Source-Bibliothek orientiert hat.

Architektur- und Managementthemen

Neben den Technologien nahmen auch Architektur- und Managementthemen wieder einen prominenten Platz ein. Das äußerte sich vor allem im SOA Management Day und im BPM Day, die die Nachfolge des Management Day auf der Konferenz angetreten haben. Bei beiden handelt es sich um ganztägige Tracks, die gut besucht waren.

Das Programm des SOA Management Day spiegelt wieder, dass sich bei diesem Thema der Fokus inzwischen von technischen Problemen – SOAP- und WSDL-Bibliotheken, deren Handhabung, Interoperabilität, nichtfunktionale Eigenschaften etc. – hin zu organisatorischen Themen verschoben hat. Dabei ist ein wichtiger Faktor, dass SOA fast per definitionem nicht Sache einzelner Projekte ist, sondern es hier um das Zusammenspiel mehrerer Systeme geht. Und weil das einheitlich geschehen soll, gelangt man schnell zu übergreifenden Themen wie der unternehmensweiten Einführung einer Technologie oder einheitlicher Tools und Standards. So behandelte ein Vortrag z.B. die Frage: „Wo lohnt sich SOA?“ Das Schlagwort SOA wird nach wie vor stark als Verkaufsargument von Tool-Herstellern verwendet, was erheblich zum Hype um dieses Thema beiträgt. Der Vortrag zeigte jenseits leerer Worthülsen die archi-

tektonischen Prinzipien hinter SOA auf und diskutierte auf dieser Basis Use Cases, in denen es echten Nutzen bringt.

Eine andere Session behandelte das Thema „SOA Governance: Repositories, Registries und das Management großer Systeme“. Beim Einsatz von SOA in großen Systemlandschaften – und dafür wurde es ja schließlich entworfen – ist eine Verwaltung der Services in einem zentralen Repository erforderlich, wenn man den Überblick behalten und Konsistenz sicherstellen will. Die Entscheidung für ein Produkt, das diese zentrale Verwaltung übernimmt, hat jedoch weitreichende Auswirkungen und sollte deswegen wohlüberlegt erfolgen.

Der BPM Day betrachtete die Integration von Anwendungslandschaften aus einem anderen Blickwinkel, nämlich demjenigen der Geschäftsprozessmodellierung. Auch diese Domäne ist in der letzten Zeit erheblich gereift, und eine zunehmende Zahl an Projekten setzt auf eine Prozess-Engine als Bestandteil ihrer Middleware auf.

Auch bei diesem Thema lag der Schwerpunkt stark auf BPM als unternehmensweitem Thema. Zu den Standardisierungs- und Koordinationsthemen gesellt sich hier noch, dass die modellierten Prozesse nicht nur Maschinen, sondern auch Menschen betreffen und diese einbezogen werden müssen. An dieser Stelle ergeben

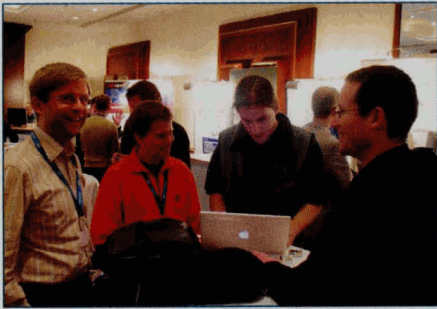


Abb. 4: Die vier W-JAX-Speaker Jürgen Höller, Bernd Kolb, Eberhard Wolff und Martin Marinschek auf frischer Tat ertappt

sich deutliche Berührungspunkte mit klassischer Unternehmensberatungstätigkeit.

Modellieren und Generieren

Auch das Thema „Modellieren und Generieren“ war wieder auf der Konferenz vertreten. Hier lagen die thematischen Schwerpunkte ebenfalls nicht mehr auf den technologischen Grundlagen. Ein Fokus war dabei die Verwendung modellgetriebener Technologien im Kontext konkreter Architekturen, vor allem serviceorientierter Architekturen. Dabei geht es um Schnittstellen, die Datentypen verwenden. Es gibt Komponenten, die Schnittstellen bereitstellen und andere Schnittstellen benötigen. Die Komponenten laufen in unterschiedlichen Umgebungen auf unterschiedlicher Hardware. Das Ganze soll zentral verwaltet und koordiniert werden, und bei dem Ganzen gibt es eine Fülle an technischem Infrastruktur-Code, der im Prinzip überall gleich aussieht: Dies ist ein natürlicher Anwendungsfall für modellgetriebene Entwicklung.

Daneben gab es auch Vorträge über fortgeschrittenere Techniken im Zusammenhang mit MDSD. So z.B. eine zweiteilige Session über Modelltransformationen. Das Hauptziel modellgetriebener Entwicklung besteht darin, auf einer höheren Abstraktionsebene zu entwickeln, dabei Architekturentscheidungen festzulegen und zu vereinheitlichen. Dieser Sprung von der Abstraktionsebene des Modells zu derjenigen des generierten Codes erfolgt traditionell vollständig in den Generierungs-Templates.

Das führt zu monolithischen Generatoren und behindert die Wiederverwendung. Abhilfe schaffen hier Modelltransformationen, mit denen man den Sprung

zwischen den Abstraktionsebenen in mehrere kleine Schritte aufteilen kann. Inzwischen gibt es eine Reihe von Werkzeugen, die beim Aufsetzen solcher Transformationen helfen, und sie sind so weit ausgereift, dass ein Vergleich sinnvoll ist. Die thematische Schwerpunktsetzung zeigte, dass MDSD als Thema inzwischen „angekommen“ ist und eine gewisse Reife und Stabilität jenseits eines Hypes erreicht hat.

Ercatons

Ein persönliches Highlight der Konferenz war für mich ein Vortrag, auf den ich eher zufällig aufmerksam wurde: „Ercatons und Organisches Programmieren mit Java“. Das klingt ein wenig nach einer Kreuzung aus griechischer Mythologie und Biobauer, ist aber ein umfangreicher Ansatz beim Programmieren mit dazugehörigem Framework, der anders ist – jedenfalls anders als alles, was einem sonst so begegnet.

Die Idee hinter diesem Ansatz ist, dass ein großer Teil der Komplexität beim Programmieren künstlich ist und sich vermeiden lässt, wenn Programme „organisch“ wachsen können, statt dass sie am grünen Tisch konzipiert werden müssen. Alles soll ein „Ding“ sein – der Fachbegriff dazu, der zugebenermaßen besser klingt, ist „Ercaton“ – und sowohl die Daten als auch die Funktionalität enthalten, wobei man dabei möglichst wenig Quelltext schreiben müssen soll. Der Ansatz vereint eine Fülle an innovativen Ideen und Konzepten, die zwar für sich genommen bekannt sind, hier jedoch in einer neuen Weise zusammengestellt sind.

Das Herzstück des Ansatzes ist die Ercato-Engine, die als Application Server läuft und in der die Ercatons „leben“. Ercatons sind einfach XML-Dokumente, die sowohl Daten als auch Funktionalität enthalten, Letzteres durch Verweise auf deployten Code in einer normalen Programmiersprache wie z.B. Java. Ungewöhnlich ist dabei, dass es keine „Klassen“ oder sonstige Typen gibt, sondern nur Objekte – eben die Ercatons. Duplizierung kann man dabei dadurch vermeiden, dass ein Ercaton auf ein anderes als seinen Prototypen verweist und von ihm – auf Instanzebene! – erbt. Diese Art von Objekten ohne Klassen ist an Sprachen wie Self angelehnt.

Getreu dem Motto, dass eine Anwendung möglichst wenig Code enthalten soll, stellt die Ercato-Engine von Haus aus eine Webschnittstelle zur Verfügung, um Ercatons anzuzeigen, zu suchen und zu verändern. So kann ein Anwender unmittelbar mit den Objekten interagieren, was Parallelen zum Naked-Objects-Ansatz hat. Operationen, die das Ercaton bereitstellt, stehen dabei als Buttons zur Verfügung. Und das Ändern von Ercatons – sowohl das Bearbeiten von Daten als auch z.B. das Hinzufügen von Operationen – erfolgt zur Laufzeit transaktionsgesichert.

Eine umfassende Beschreibung des Ansatzes würde den Rahmen dieses Konferenzberichts sprengen, aber der Ansatz verdient die weitere Beobachtung. Auf den ersten Blick scheint es sich um eine exotische Nische zu handeln, die nicht praxistauglich ist. Aber es sind immerhin schon unternehmenskritische Anwendungen damit entwickelt worden. Der Ansatz ist auf jeden Fall erfrischend anders und erweitert den Horizont und das Denken.

Das Framework ist zurzeit nicht frei verfügbar, was die Bedeutung des ganzen Ansatzes erheblich einschränkt. Die Hersteller überlegen allerdings, das Framework als Open Source zu veröffentlichen, und wenn das geschieht, lohnt sich sicher ein näherer Blick darauf.

Fazit

Die W-JAX hatte auch dieses Jahr wieder eine Fülle interessanter Vorträge mit einer großen thematischen und stilistischen Bandbreite. Ein Schwerpunkt der Konferenz lag wieder auf technischen Themen und aktuellen APIs. Daneben spielten aber Prozess- und Management-Themen eine wichtige Rolle, was eine zunehmende Reifung der Softwareentwicklung allgemein und von Java als Plattform im Besonderen widerspiegelt.

Das Ganze hat auf jeden Fall viel Spaß gemacht, und ich habe fest vor, nächstes Jahr wieder dabei zu sein.

Termine

- JAX 07, EAKon und Eclipse Forum Europe 07: 23. bis 27. April 2007, Wiesbaden
- W-JAX 07 und EAKon: 12. bis 16. November 2007, München