

Info Bridge

Herausgeber: Object Engineering GmbH • www.objeng.ch • Juli 2007 • Nr. 8

Editorial

Liebe Leserinnen, liebe Leser



Die Softwareentwicklungskultur von heute, im speziellen für Inhouse-Projekte, folgt leider nicht immer dem Software-Engineering-Gedanken. Dass dieses kurzfristige Denken nicht zu einer qualitativ, robusten

und flexiblen IT-Basis führt, welche keine Folgekosten nach sich ziehen wird, lässt sich leicht erahnen.

Flexible Plattformen sollten wir haben, damit die laufenden und zukünftigen Anforderungen der Unternehmung erfüllt werden können.

Ein weiser Unternehmer legt in guten Zeiten etwas auf die Seite, um es zur Not oder bei Unvorhergesehenem zu verwenden.

Ein Hauseigentümer macht dasselbe, damit er das Haus instand halten kann, wobei er sich bewusst ist, dass nicht nur Wetter, Sonne und Gebrauch dem Zustand des Hauses abträglich sind. Auch neue Anforderungen durch Gesetz oder Mieter müssen erfüllt werden können.

Klar sichtbar ist dies bei der Dorfplanung. Ein gutes Verkehrsnetz instand gehalten und den Bedürfnissen angepasst, ergibt einen reibungslosen und flüssigen Verkehr, der auch für die Anwohner erträglich ist. Gerade in diesem Umfeld gilt es Voraussicht walten zu lassen, was bei IT-Umgebungen nicht weniger zu empfehlen wäre.

Bei der Unternehmens-IT kann Flexibilität, Robustheit und Geschwindigkeit den Unternehmens-Gewinn sichern, ja zum Ansteigen bringen. Während viele über die heutige Hardware-Entwicklung staunen und diese bewundern, weist die Software-Entwicklung wenig oder keine Fortschritte mehr auf. Nicht dass man täglich neue Entwicklungsmethoden verwenden sollte, im Gegenteil, man sollte bewährte

Das sagen unsere Kunden:

Fischer AG Herzogenbuchsee



Unsere Firma ist weltweit führender Hersteller von Präzisions-Spindeln, welche in Hochgeschwindigkeits-Bearbeitungszentren und Schleifmaschinen eingesetzt werden, wo kürzeste Bearbeitungszeiten gefordert sind.

Für die Verwaltung dieser sehr speziellen Komponenten haben wir selbst eine eigene Software-Anwendung, basierend auf einer mySQL-Datenbank und jsp-Applikation-Server (tomcat), entwickelt. Dieser Ansatz wurde gewählt, damit das weltweit wachsende Service-Stellennetz jederzeitigen Zugang zu

Methoden und Systematiken im-Gange halten und sicherstellen, dass die Mitarbeiter sie beherrschen. Dies wurde und wird in anderen Industriezweigen gemacht, um zu verhindern, dass ein Unternehmen plötzlich nicht mehr konkurrieren kann. Es geht hier nicht darum, alles in einen Topf zu werfen, aber es ist oft zu beobachten, dass Coding und Scripting ohne Design eine vorherrschende Rolle einnehmen. Architektur und Design ist nur bei wenigen Profis anzutreffen. Und immer gibt es "gute Gründe" dafür, es nicht zu tun.

Stellen Sie sich zur Probe doch mal folgende Frage: *Würden Sie in ein Flugzeug steigen und damit an eine entfernte Destination fliegen, welches von einer Software gesteuert wird, die ohne Methodik entworfen und in fragwürdiger Qualität realisiert ist, ähnlich der Software, die Sie täglich antreffen?* Ich hoffe, dass es Ihnen dabei nicht die Nackenhaare aufstellt und kalt den Rücken runter läuft. Guten Flug!

Mit freundlichen Grüßen

Ihr *Andreas Koch*

diesen wichtigen Informationen hat und Einträge direkt vornehmen kann. Mit der Markt-Expansion stiegen die Anforderungen an diese Spindelverwaltung. Für die Weiterentwicklung fehlten interne Ressourcen und ein externer Partner war gefragt.

Object Engineering GmbH konnte uns in einem Support-Fall rasch und unkompliziert weiterhelfen. Da war es naheliegend, diesen Partner mit der Weiterentwicklung zu betrauen. Über die letzten drei Jahre hinweg konnten wir mit Object Engineering diese Spindelverwaltung in mehreren Etappen bedarfsgerecht ausbauen. Jede Ausbau-Stufe konnte unverzüglich in Produktion genommen werden und der nächste Ausbau konnte entsprechend gut vorbereitet werden.

Die Applikation konnte mittels Open-Source Komponenten kostengünstig erweitert werden. So wurden für die PDF-Generierung Jasper-Reports und iText eingesetzt. Die Verwendung einer Reporting-Lösung hat den Vorteil, dass das Layout der generierten Dokumente ohne Programm-Anpassungen und ohne Programmierkenntnisse verändert werden kann.



Der Einsatz von Apache Struts gewährleistet, dass die hohe Komplexität der Applikation im Griff ist, und dass die Applikation immer noch übersichtlich und wartbar ist.

Mit der Erweiterung dieser für uns wichtigen Applikation verfügen wir und unsere Kunden über eine wesentlich verbesserte Qualität der Daten. Dies trägt dazu bei, dass die Qualität und Zuverlässigkeit unserer Spindeln und Komponenten weiter gesteigert werden kann.

Beat Burkhard

Leiter CAD Systeme
Fischer AG, Herzogenbuchsee

Software-Brückenpfeiler für tragfähige Brücken

SOA (Service Orientierte Architektur) als Konzept wird sich für die nächsten fünf bis zehn Jahre behaupten können, wobei es wahrscheinlich immer wieder unterschiedlich vermarktet werden wird. Sicher werden noch einige "Überraschungen" mit den hochgepriesenen Technologien, wie beispielsweise mit Web-Services, zu erwarten sein. Auch wird der Bereich der IT-Unterstützung von Geschäftsprozessen wesentlich höhere Anforderungen an die Beteiligten stellen als gemeinhin angenommen wird. Soll die IT-Architektur trotzdem den laufend und immer schneller ändernden Anforderungen gerecht werden, so muss diese auf ein robustes Fundament gestellt werden.



Einleitung

Je nach Sichtweise unterteilt man eine IT-Architektur nach serviceorientiertem Muster in drei bis vier Schichten:

- **Enterprise oder Application Frontend Layer**
- **Business Processing Layer**
- **Intermediary Layer**
- **Service Provider Layer**

Jeder Layer hat ein relativ genau umschriebenes Aufgabenpaket zu erfüllen. Würde man versuchen alle Schichten durchgängig auf einen Schlag zu realisieren, dann würde der Erfolg wohl ausbleiben, denn man käme kaum auf einen grünen Zweig. Wenn SOA für ein Unternehmen zum Thema wird, sind in praktisch jedem Fall viele bis sehr viele Systeme, Applikationen sowie entsprechend komplexe Integrationspfade im Spiel. Diese folgen in der Regel keiner durchgängigen Struktur, was ein Umsetzen in eine SOA zur Herausforderung werden lässt.

Darum ist man bei IT-Architektur gut beraten, wenn man versucht, die SOA-Zielarchitektur auf allen Ebenen anzustreben, aber zuerst auf der Service-Provider-Ebene eine gute Basis schafft.

Robuste Brückenpfeiler

Die Service-Provider-Ebene ist neben der Infrastruktur (Service-Bus etc.) eine fundamentale Stütze. Diese muss aus robusten und normierten Komponenten bestehen, welche unabhängig von Technologie-Änderungen die Investitionen in die Zukunft sichern.

Dabei gilt:

- **Service-Anforderungen und -Identifikation**
- **Generelle und flexible Schnittstelle definieren**
- **Contract beschreiben**

Jeder Service muss identifiziert und definiert sein, sonst kann er auf keiner anderen Ebene verwendet werden. Die Anforderungen müssen klar aber trotzdem breit spezifiziert werden. Das heisst, dass sich darin nicht spezifische Business-Anforderungen wieder spiegeln sollten, sondern grundlegende Funktionen festgelegt werden.

Weiter muss einerseits eine klare Schnittstellen-Doktrin in der Architektur festgelegt werden, und die bereitgestellten Services müssen in einem Verzeichnis (Repository) den Prozess- und Applikations-Architekten zur Verfügung stehen. Sie bilden nämlich die Grundlage für Durchgängigkeit und Langlebigkeit der darauf aufgebauten Lösungen.

Dass Services professionell und robust beschaffen sein sollten, könnte man meinen, sei klar. Dies kann aber nicht oft genug wiederholt werden. Es sollte die Aufgabe ihres besten Entwicklerteams sein, solche Services zu entwickeln.

Auf der Seite der Consumer müssen die Services mit gut verständlichen und einfach anwendbaren Schnittstellen ausgerüstet sein. Hier kommen Standard-Technologien zum Zug, doch Technologie alleine ist noch kein Garant für Qualität. Schnittstellen müssen durchdacht und sorgfältig entworfen werden.

Die Konstruktion der Services ist die Aufgabe der Software-Engineers. Folgende Unterteilung ist eine Sicht und soll für die Einteilung verschiedenartiger Services innerhalb der Architektur helfen. Es soll hier aber erwähnt werden, dass Services auf jeder Schicht einer Service-Orientierten-Architektur

und nicht nur auf der Service-Provider-Schicht vorkommen können.

Facade-Services (Enterprise Layer)

Diese Services sind die Gateways zur Organisation (B2B) und können, im Gegensatz zu den unten beschriebenen, als Webservices ausgelegt werden. Typischerweise passt sich ein solcher Service den Anforderungen und gewünschten Schnittstellen der externen Konsumenten an und transformiert die Daten von und in die internen Strukturen. Pro Consumer darf es hier einen eigenen Service geben. Es ist die Fassade davon.

Intermediary-Services (Intermediary Layer)

Diese Services haben eine gewisse technische Aufgabe und stellen vom Business-Prozess geforderte Funktionalitäten "mundgerecht" bereit. Sie können mehrere Aufrufe zu Basis-Services kapseln und quasi als Container eine gewisse Transaktions-Funktionalität übernehmen.

Basis-Services (Intermediary Layer)

Diese Services werden typischerweise neu erstellt und decken Basis-Funktionalitäten neuer Business-Logik ab. Sie werden aufgrund von neuen Anforderungen erstellt und stehen als eigenständige Services oder als Komponenten (EJB) innerhalb einer Application-Server-Plattform zur Verfügung.

Hier ist die Granularität und das logische Design der Schnittstelle entscheidend.

Wrapped-Services (sowohl Intermediary Layer wie auch Service Provider Layer)

Gekapselte Services werden vor allem zu Beginn einer Migration in eine service-orientierte Umgebung am häufigsten anzutreffen sein.

Nach wie vor haben Unternehmen bewährte Applikationen, die qualitativ und vom Business-Know-how her nicht zu unterschätzende Ressourcen bilden. Durch eine logisch geeignete Fassade (Wrapper) wird nur gerade die Funktionalität nach aussen publiziert, welche aufgrund der Business-Anforderungen auch wirklich gewünscht ist. Da solche gestandenen Applikationen auf der

Services: die Pfeiler von SOA

Kehrseite der Medaille oft ein Wartungsproblem darstellen, kann dank einem solchen Wrapper hinter der Fassade schrittweise eine Erneuerung stattfinden, ohne dass die Konsumenten davon betroffen sind. Zu den Wartungsproblemen gehören unter anderem nicht mehr unterstützte Technologie, Know-how-Mangel, wie zum Beispiel bei Entwicklern, die mit älteren Sprachen nicht mehr umgehen können und vieles mehr.



Inzwischen sind nicht nur Programme in Sprachen wie Cobol, Fortran, PL/1 und ähnlichen als Legacy zu bezeichnen. Auch Programme in C, C++, ja unter Umständen sogar in Java, gehören, wenn in die Jahre gekommen, bereits zu den Legacy-Systemen. Ein Wrapping kann bei älteren Technologien dem Refactoring (bei neueren Technologien) vorgezogen werden oder auch als Zwischenschritt zum Gewinnen von Zeit gewählt werden. In der Regel und mit etwas Erfahrung lässt es sich technisch einfach realisieren, je nachdem wie systematisch die Applikation beschaffen ist.

COTS-Services

Heute gibt es noch keine oder nur wenige Komponenten auf dem Markt, die auf der Granularität von logischen Services eingekauft werden können. Leider herrscht eine falsche Meinung vor, man könne sogenannte COTS (Commercially Off The Shelf)-Applikationen fertig einkaufen und damit die meisten Probleme lösen. Dabei wird oft vergessen, dass gerade diese COTS-Systeme meist monolytische Komponenten sind, die das benötigte Business-Modell und -Funktion in der Regel nicht abdecken. Zudem sind sie technologisch auch zu den Legacy-Systemen zu zählen und die Integration kostet bald mehr als eine neu erstellte Komponente.

Es ist sehr empfehlenswert, dass solche COTS-Systeme, genauso wie bestehende Legacy-Systeme, mit einer kapselnden Fassade versehen werden. Diese Fassade

stellt, wie bei Legacy-Services, nur die Funktionalitäten zur Verfügung, welche von den Business-Prozessen her auch benötigt werden. Dadurch hat man bereits vorgesorgt, dass man die Hersteller-Abhängigkeit minimieren kann und zu einem späteren Zeitpunkt nicht blockiert ist, flexibel und innovativ auf neue Anforderungen zu reagieren. Der Businessvertreter wird jedem IT-Architekten dankbar sein, der dies beachtet hat, obwohl er in der Regel nicht bereit ist, die zusätzlichen Kosten vorher zu investieren. Diese Investition lohnt sich jedoch auf jeden Fall. Beim Abwägen, ob mit COTS oder nicht, ist ein Faktor entscheidend, nämlich ob es nicht sinnvoller wäre, zugeschnittene Komponenten von Anfang an zu konstruieren.

Service-Design

Das Design von Komponenten ist eine entscheidende Sache, denn wenn die einzelnen Bausteine (Brückenpfeiler) nichts taugen, wie soll dann damit ein IT-Gebäude längerfristig standhalten? Applikations-Architektur und Schnittstellen-Design bilden die Grundlage dazu. Arbeitet man mit bewährten Frameworks und der geeigneten Technologie und baut daraus die Komponenten (Services) auf, kann Know-how und Code wieder verwendet werden.

Software-Engineering bei der Realisierung von Services

Unsere langjährige Erfahrung beim Bau von verteilten Software-Systemen hat klar gezeigt, dass sorgfältig entworfene und qualitativ hochstehend konstruierte Komponenten für Langlebigkeit und Robustheit des gesamten Systems garantieren. So ersparen wir dem Kunden unnötige Kosten und man kann neue Projekte in Angriff nehmen. Wenn eine SOA umgesetzt werden soll, dann darf nicht jeder Service auf eine andere Form entwickelt werden. Eine vernünftige Normierung ist angebracht, damit effizient Services entwickelt und auch über eine längere Zeit ohne grossen Aufwand gewartet werden können. Hier sind die Software-IngenieureInnen mit Architektur-Erfahrung gefragt.

Autor: Andres Koch,
dipl. El. Ing. HTL, M. Math (CS)

Die "alten" Systeme und Applikationen ablösen ...



... bevor sie unter "Heimatschutz" gestellt werden!

Auch wenn mit dem Millenniums-Wechsel viel Boden gut gemacht wurde, läuft die Zeit weiter.

Gestandene Applikationen mögen zwar noch gut funktionieren, nur steigen die Wartungskosten in unerwünschte Höhen.

Wir haben langjährige Erfahrung bei der Renovation und Wartung von Legacy-Applikationen.

Wir helfen Ihnen beim Ablösen oder Erreichen der SOA-Fähigkeit beliebiger Applikationen und Systeme:

- Analyse des Bestehenden
- (Ab-) Lösungskonzept dafür
- Redesign und Architektur-Korrekturen
- Migration von C++, C und anderen Sprachen und Technologien
- Ersetzen oder Re-Factoring in C++, C und Java
- Umsetzung parallel zum laufenden Betrieb
- Regresstest-Hilfen
- Integration und Test
- Outsourcing der Wartung der renovierten Applikationen

Kontaktieren Sie uns für ein kostenloses und unverbindliches Assessment.

Danach sind Sie **in der Lage zu entscheiden**, ob Sie lieber in den "Heimatschutz" oder in die Zukunft investieren:

E-Mail: tech@objeng.ch

Aktuelles

Brückenwettbewerb

Wer hätte nicht Lust entlang dieses Flusses zu wandern, ins kühle Nass des Wassers bei dieser Brücke zu steigen, sich auf den Liegebetten aus Stein am Ufer zu sonnen....



Die Wettbewerbsfrage lautet:

Was ist der Name dieser alten Steinbrücke?

Ihre Lösungsantwort senden Sie bitte bis zum 31. August 2007 an:

E-Mail: infobridge@objeng.ch

Fax: **044 400 47 07**

Unter den Teilnehmern mit der richtigen Antwort wird eine passende Überraschung ausgelost.

Viel Spass und vielleicht zieht es Sie ja in den Ferien in diese wunderschöne Gegend.

Wettbewerbs-Auflösung

Die Auflösung zum letzten Wettbewerb in der InfoBridge 7 ist: Die Kettenbrücke (ungarisch Széchenyi Lánchíd) überspannt die Donau und verbindet die beiden Stadtteile **Buda** und **Pest** — in Budapest. Der Gewinner eines Reisegutscheins heisst Herr K. Bucher

Herzliche Gratulation!



Unsere Mitarbeiterin: Simone Califano



Die Stimme am Telefon in der Object Engineering gehört (in der Regel) Simone Califano. Sie ist seit bald zwei Jahren für unsere Administration und für die reibungslose Kommunikation innerhalb und ausserhalb der Firma verantwortlich. Zudem kümmert sie sich um unsere Entwickler, damit diese ungestört und gut mit Kaffee versorgt arbeiten können und das Mittagessen trotzdem nicht verpassen.

In der Freizeit tanzt Simone Califano fürs Leben gern, was sie mit professioneller Routine angeht.

Amazonas Workflow V 2.3

Seit Mai 2007 ist die neue Version 2.3.0 der Amazonas Workflow Engine verfügbar. Neben diversen Verbesserungen bestehender Funktionalität und Bugfixes bringt sie folgende neue Features mit:

- Neue Funktionen der AWF-Skript Sprache
- Flexiblere Datenbank-Konfiguration
- Unterstützung von Oracle als Datenbank

Amazonas Workflow V 2.3

Amazonas kann mit dem folgenden Link heruntergeladen werden:
<http://web06.objeng.ch/support/downloads/awf/workflow-de.shtml>

Unser neuer Mitarbeiter: Darius Bohni



Haben Sie schon einen Blick in unsere neue Home-Page geworfen? Sie wurde von unserem neuen Mitarbeiter Darius Bohni entworfen und realisiert. Er ist seit August 2006 für die Object Engineering GmbH als Software-Entwickler tätig und ist auch verantwortlich für unseren Kundendienst.

Nach dem eidgenössischen Abschluss in Informatik, belegt er seit diesem Frühjahr das Nachdiplom-Studium in Software-Engineering an der Technischen Hochschule Rapperswil.

Er ergänzt unser Team speziell im Bereich von Benutzer-Oberflächen und graphischer Darstellung, was ihm mit seiner Leidenschaft für Ästhetik sehr gut liegt. Die Realisierung der Applikationen setzt er mit Java und PHP um. Er hat aber auch sehr gute Kenntnisse in anderen Technologien wie Javascript und Ajax, VB und ASP. Mit seinem Background in .Net und Windows-Server ergänzt er unser Integrations-Team, wenn es darum geht, „Brücken“ zu Systemen der „anderen“ Welt zu schlagen. Natürlich kennt er sich auf verschiedenen Linux-Plattformen aus, was bei uns eine Vorbedingung ist.

Object Engineering GmbH

Birmensdorferstr. 32
CH-8142 Uitikon-Waldegg

Tel: +41 (0) 44-400 47 00
Fax: +41 (0) 44-400 47 07

www.objeng.ch
info@objeng.ch



Member of the
Solution Network Group

Object Engineering® ist ein eingetragenes
Warenzeichen im Besitz der
Object Engineering GmbH