

HOCHSCHULE HEILBRONN
Hochschule für Technik und Wirtschaft

Studiengang Electronic Business (EB)

Diplomarbeit

**Konzeption und partielle Umsetzung
der Umstellung einer Webanwendung von
Applets auf Applikation**

vorgelegt bei
Professor Dr. Gröschel

von
Andrea Fuhrmann
Matr.-Nr. 157168

im

WINTERSEMESTER 2005/2006

Inhaltsverzeichnis

Deckblatt.....	I
Sperrvermerk.....	II
Eidesstattliche Erklärung.....	III
Inhaltsverzeichnis.....	IV
Abkürzungsverzeichnis.....	VII
Abbildungsverzeichnis.....	IX
Tabellenverzeichnis.....	XI
Management Summary.....	XII
1 Einleitung.....	1
1.1 Problemstellung	1
1.2 Aufbau der Arbeit	1
2 Einführung in Global Ordering.....	3
2.1 Allgemein.....	3
2.2 Systemüberblick.....	4
2.3 Das GO-Portal.....	5
2.4 Intranet-, Extranet- und Internet-Anbindung	8
3 Aufgabenstellung	11
3.1 Ausgangssituation.....	11
3.1.1 Signierung des Applets	11
3.1.2 Heterogene Systemvoraussetzungen.....	12
3.1.3 Java Plug-in bzw. Java Laufzeitumgebung.....	12
3.1.4 Hoher Migrationsaufwand	13
3.1.5 Lösungsansatz: Applikation.....	14
3.2 Anforderungsanalyse	14
3.3 Aufgabenstellung	15

4 Entwurf der technischen Architektur	18
4.1 Darstellung der Ist-Architektur.....	18
4.1.1 Analyse der Ist-Architektur.....	18
4.1.1.1 Das GO-Applet.....	19
4.1.1.2 Der GO-Caching Mechanismus.....	22
4.2 Bewertung initialer Start- und Lademechanismen.....	24
4.2.1 Java Web Start.....	24
4.2.1.1 Installation und Nutzung.....	25
4.2.1.2 Technik und Funktionsweise.....	26
4.2.2 StarterApplet.....	28
4.2.2.1 Installations- und Startprozess.....	28
4.2.2.2 Applikationsdeskriptor und ScriptEditor.....	30
4.2.2.2.1 Initiale Variante des ScriptEditors.....	30
4.2.2.2.2 Update Variante des ScriptEditors.....	32
4.2.2.3 Technik und Funktionsweise.....	33
4.2.3 Anforderungskatalog.....	35
4.2.4 Bewertung und Produktentscheidung.....	36
4.3 Aufstellung der Soll-Architektur.....	39
5 Erstellung eines Prototyps	41
5.1 Vorgehensweise.....	41
5.2 Erstellung applicationrunner.jar.....	41
5.2.1 Apache Ant.....	43
5.2.2 Die Konfigurationsdatei applicationrunner.properties.....	45
5.3 Bereitstellung der Applikation.....	46
5.3.1 Einbettung des StarterApplets.....	47
5.3.2 Verzeichnisstruktur des Webservers.....	48
5.3.3 Erstellung des Applikationsdeskriptors.....	49
5.4 Einbinden der Java Laufzeitumgebung.....	51
5.5 Problem der Mehrsprachigkeit.....	53
5.6 Zugriff über Reverse Proxy.....	55

5.7 Hilfeseiten integrieren.....	56
6 Performance-Messungen.....	58
6.1 Zielsetzung und Vorgehensweise	58
6.2 Messbare Aktionen	58
6.2.1 Initialer Aufruf.....	58
6.2.2 Wiederholte Aufrufe	59
6.2.3 Aufrufe spezieller Dialoge.....	59
6.3 Übersicht über die Messergebnisse.....	60
6.4 Schlussfolgerungen.....	61
7 Strategien zum Roll-out.....	63
7.1 Roll-out Management	63
7.1.1 Stichtagsumstellung	64
7.1.2 Stufenkonzept	65
7.1.3 Parallelbetrieb	67
7.2 Konzept zum Roll-out von GO.....	67
7.2.1 Kenntnis über die Zielumgebungen.....	67
7.2.2 Einteilung in Roll-out-Gruppen.....	69
7.2.3 Information der Anwender.....	69
7.2.4 Verteilung der Applikation	70
7.2.5 Bereitstellungs- und Testphase	70
7.2.6 Produktivsetzung	70
7.2.7 Notfallplan und Rückfallstrategie	71
8 Fazit.....	73
8.1 Zusammenfassung	73
8.2 Kritik.....	74
8.3 Ausblick.....	75
Anhang.....	XVI
Literaturverzeichnis.....	XXXI

Management Summary

In vielen Konzernen existieren Anwendungen, die weltweit von Mitarbeitern und sonstigen Benutzergruppen wie beispielsweise Zulieferer genutzt werden müssen. Um Zugriff auf solche Systeme zu erhalten, wird oftmals die Internettechnologie genutzt. Jedoch ist es gerade in internationalen Unternehmen schwierig, gleiche Voraussetzungen bezüglich Hardware oder Software zu schaffen. Dies ist zum einen zurückzuführen auf das unerschöpfliche und stets wachsende Angebot und zum anderen auf länderspezifische Standards. Da für jedes System bestimmte Voraussetzungen zur Nutzung erfüllt werden müssen, wird es den Mitarbeitern bei unterschiedlichen Systemen erschwert, ihre Arbeit ordnungsgemäß zu erfüllen.

Auch bei der DaimlerChrysler AG existieren Anwendungen, die weltweit den Mitarbeitern verfügbar gemacht werden müssen. Eine dieser Anwendungen ist Global Ordering (GO). GO ist die informationstechnische Umsetzung des Kundenauftragsprozess der Mercedes Car Group innerhalb der DaimlerChrysler AG. Dieser Prozess wird durchgängig von der Absatz- und Programmplanung über die Bestellung durch den Kunden bis hin zur Übergabe des Fahrzeugs an den Kunden abgebildet. Im Hintergrund dieser Anwendung steht eine Datenbank, in der alle Daten von Vertrieb und Produktion gespeichert werden. GO ist realisiert als ein Client-Server-System, das auf einem Großrechner angesiedelt ist. Der Anwender nutzt die Dialoge über Applets, die über das Internet verfügbar sind. Um die Kommunikation zum Server zu realisieren, sendet der Client `ServerRequests` an ein `Server`-Objekt, das wiederum die Anfragen an den Host weiterleitet. Als Antwort erhält der Client von dem `Server`-Objekt ebenfalls einen `ServerRequest`. GO ist über das Intranet, das Extranet und das Internet zugänglich. Anwender, die dem DaimlerChrysler Corporate Network (DCCN) angehören, greifen über das Intranet zu. Da nicht alle Benutzergruppen Zugriff auf das DCCN haben, muss der Zugriff hier anderweitig realisiert werden. In Ländern, in denen eine Extranet-Anbindung an das DCCN existiert, werden die Anwender über diese Art angebunden. Existieren keine Extranet-Anbindungen, so erfolgt der Zugriff über das Internet.

Als die Entscheidung für eine Realisierung mittels Applets fiel, wurde der größte Vorteil dieser Technologie in der installationsfreien Ausführbarkeit der GO-Dialoge gesehen. Durch den weltweiten Einsatz ergaben sich allerdings auch Nachteile, die sich mit der Nutzung immer mehr heraus kristallisierten:

- Signierung des Applets

- Heterogene Systemvoraussetzungen
- Java Plug-In bzw. Java Laufzeitumgebung
- Hoher Migrationsaufwand.

Durch die Signierung des Applets fallen ständig wiederkehrende Kosten an. GO ist weltweit im Einsatz und nicht alle Anwender sind in das DCCN integriert. Daher ist es fast unmöglich gleiche Systemvoraussetzungen zu schaffen. Verschiedene Länder haben unterschiedliche Betriebssysteme, Browsersoftware sowie Java Plug-Ins im Einsatz. Jedoch ist es für die Nutzung von GO erforderlich, dass hier eine gewisse Homogenität bei den Anwendern vorhanden ist. Erfüllen die Anwender die Nutzungsbestimmungen nicht, so treten Fehler bei der Nutzung auf und GO ist im schlimmsten Fall nicht aufrufbar. Der höchste Aufwand hinsichtlich Kosten fällt jedoch bei einer Migration auf eine nächst höhere Java Laufzeitumgebung an. Um diese Kosten einzusparen und unabhängiger von äußeren Einflüssen zu sein, fiel die Entscheidung eine Studie bezüglich der Umstellung von GO auf eine Applikation durchzuführen. In der sich anschließenden Anforderungsanalyse wurde diese Entscheidung bestärkt. Die Vorteile der Applikation werden vor allem in den o. g. Bereichen sowie der Softwareverteilung inklusive Java Laufzeitumgebung gesehen.

Die Grundlage der Applikationslösung bildete die Aufstellung der neuen technischen Architektur. Um diese aufzustellen, wurde zunächst die Ist-Architektur detailliert analysiert. Dazu wurde der Aufbau der GO-Einstiegsseite untersucht. Vor der Aufstellung der Soll-Architektur wurden Mechanismen überprüft, mit denen ein initialer Start der Applikation vollzogen werden kann. Zur Auswahl standen Java Web Start von Sun Microsystems und das StarterApplet der DaimlerChrysler AG.

Java Web Start ist ab Version 1.4 ein Bestandteil der Java Laufzeitumgebung oder des Java Developer Kit. Die Technologie ermöglicht es, Java Programme über das Internet oder andere Netzwerke zu verteilen und in einem einzigen Schritt herunterzuladen sowie zu starten. Ein StarterApplet ist eine Eigenentwicklung der DaimlerChrysler AG. Es wird als Java Applet in eine HTML-Seite eingebunden und startet Java Applikationen aus dem Browser heraus. Java Web Start und das StarterApplet wurden auf ihre Realisierbarkeit hin überprüft und anhand definierter Anforderungen miteinander verglichen. Die Entscheidung fiel auf das StarterApplet, da hier vor allem eine geringere Abhängigkeit von Sun Microsystems erwartet wird.

Um die Tragbarkeit der aufgestellten Soll-Architektur sowie die Entscheidung für das StarterApplet aufzuzeigen, wurde ein Prototyp erstellt. Hierbei handelt es sich um eine Applikation, die beliebige Applikationen starten kann. Die Applikation wird den Anwendern über eine HTML-Einstiegsseite zugänglich gemacht. Das StarterApplet wird in diese Seite eingebettet und sorgt dafür, dass die notwendigen Dateien auf den Client des Anwenders geladen werden. Da einer der Hauptvorteile der Applikation für die DaimlerChrysler AG die integrierte Java Laufzeitumgebung ist, wird neben der Applikation ebenfalls die JRE 1.4.2_07 auf den Client geladen. GO steht bisher in sechs Sprachen zur Verfügung, daher wurde ebenfalls auf eine Realisierung der Mehrsprachigkeit eingegangen.

Da sich GO mit der Umstellung auf eine Applikation hinsichtlich Performance und Leistungsfähigkeit nicht verschlechtern soll, wurde der Prototyp nach seiner Erstellung auf einem Testserver verfügbar gemacht und es wurden Performance-Messungen durchgeführt. Der initiale Aufruf, wiederholte Aufrufe und Aufrufe spezieller GO-Dialoge wurden jeweils auf einem Computer mit analogem Internetanschluss und einem DaimlerChrysler Standard Computer gemessen. Bei dem initialen Aufruf wird auf Seiten der Applet-Lösung simuliert, dass der Anwender bisher nicht die erforderliche Java Laufzeitumgebung auf seinem Computer installiert hat und diese daher zunächst bei Sun Microsystems herunterladen und anschließend installieren muss. Das Ergebnis der Messungen war, dass gerade bei dem initialen Aufruf der Applikationslösung Zeiteinsparungen realisiert werden.

Nach der Erstellung des Prototyps und den erfolgreich durchgeführten Performance-Messungen wurde eine Strategie für die Verteilung der Applikation definiert. Da GO weltweit im Einsatz ist, wurde für den Roll-out eine Mischvariante zwischen Stufenkonzept und Parallelbetrieb gewählt. Die Applikation wird in sechs Phasen in definierten Roll-out Gruppen eingeführt, d. h., dass die Applikation nicht überall zur gleichen Zeit verteilt wird. Anwender, die noch keinen Zugang zu der Applikation haben, werden parallel weiterhin die Applet-Lösung nutzen, bis die Applikation in allen Standorten verteilt wurde.

Abschließend lässt sich sagen, dass GO als Applikation vor allem Vorteile für die Verantwortlichen von GO mit sich bringt. Hier ist insbesondere die erzielte Unabhängigkeit von der Java Laufzeitumgebung erwähnenswert. Dadurch lässt sich der Zeitpunkt der Migration auf eine höhere Version der Java Laufzeitumgebung selbstständig und nicht fremdgetrieben bestimmen. Dies bringt im Wesentlichen erhebliche

Einsparungen in Kosten und Zeit mit sich. Auch für die Anwender bringt die Unabhängigkeit von der Java Laufzeitumgebung Vorteile. Es ist nicht mehr relevant, welche Version die Anwender auf ihrem Computer installiert haben. Durch die weitestgehende Anpassung von GO als Applikation an die bisherige Applet-Lösung wird den Anwendern die Umstellung jedoch kaum auffallen.