

ERWEITERUNG EINES GENERISCHEN FRAMEWORKS UM WEBSERVICE-FÄHIGKEIT

Entwurf und Implementierung

Bachelor-Thesis

Zur Erlangung des Hochschulgrades Bachelor of Science

an der



vorgelegt von: Frank Breitingner

Matr.-Nr. 527846

Studienbereich: Bachelor Informatik (IB)

Am: 28. Februar 2009

Betreuender Professor: Prof. Dr. Wolfgang Schramm

Korreferent: Dipl.-Inf. Stephan Gehres

Betrieb: sobedi informationsmanagement gmbh



Urheberrechte

Das vorliegende Werk ist **urheberrechtlich geschützt**.

Die vorliegende Arbeit beinhaltet interne vertrauliche Informationen der sobedi informationsmanagement gmbh. Die Weitergabe oder Nutzung des Inhalts der Arbeit im Gesamten oder in Teilen sowie das Anfertigen von Kopien oder Abschriften - auch in digitaler Form - sind grundsätzlich untersagt. Ausnahmen bedürfen der schriftlichen Genehmigung der sobedi informationsmanagement gmbh.

Erklärung zur Verwendung der Bachelor-Thesis

Hiermit erkläre ich mein Einverständnis mit den im Folgenden aufgeführten Verbreitungsformen dieser Bachelor-Thesis:

Verbreitungsform	ja	nein
Einstellung der Arbeit in der Bibliothek		X
Veröffentlichung des Titels der Arbeit im Internet	X	
Veröffentlichung der Arbeit im Internet		X

Mannheim, den 28. Februar 2009

FRANK BREITINGER

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich bei der Anfertigung dieser Bachelor-Thesis unterstützt haben.

Ein besonderer Dank gilt meinen beiden Korrektoren, Prof. Dr. Wolfgang Schramm, der mich durch seine hilfreichen Anregungen und seine Geduld immer wieder unterstützt hat, sowie Dipl.-Inf. Stephan Gehres, der mir mit seinem technischen Know-How in der Phase der Implementierung wertvolle Hinweise gegeben hat und für seine Bereitschaft, als Zweitgutachter zu fungieren.

Des weiteren möchte ich mich bei Dipl. Betriebswirt (FH) Dirk Laufer bedanken, der mich im Rahmen meiner Bachelor-Thesis bei der sobedi informationsmanagement gmbh eingestellt und mir diese Arbeit somit ermöglicht hat.

Danken möchte ich weiterhin allen Kollegen der sobedi informationsmanagement gmbh, insbesondere Alexander Pueltz, die mir die komplette Zeit mit gutem Rat beiseite standen, sowie dem Service-Team, das mir vieles erleichtert hat.

Nicht zuletzt möchte ich mich bei meinen Eltern bedanken, die mir dieses Studium überhaupt erst ermöglicht und die komplette Zeit moralisch unterstützt haben.

Inhaltsverzeichnis

Urheberrechte	I
Danksagung	II
Abbildungsverzeichnis	VI
Tabellenverzeichnis	VII
Listingsverzeichnis	VIII
1. Einführung	1
1.1. Motivation	1
1.2. Aufgabenstellung	2
1.3. Ziel der Arbeit	3
1.4. Aufbau der Arbeit	3
1.5. Firmenumfeld	4
2. Das Generic Object Framework der sobedi informationsmanagement gmbh	5
2.1. Arbeitsumgebung	5
2.2. Hochkonfigurierbare Systeme	6
2.3. Generic Object Framework	6
2.3.1. Architektur des GOF	7
2.3.2. Businessobjekte	9
3. Webservice Grundlagen	11
3.1. Service-orientierte Architektur	11
3.2. Webservice	12
3.2.1. REST	12
3.2.2. SOAP	13

3.2.2.1.	WSDL	15
3.2.2.2.	UDDI	17
3.2.3.	Bewertung der Software-Architekturstile	17
3.3.	Implementierungsmöglichkeiten für Webservices	18
3.4.	SOAP-Engine	19
3.4.1.	Anforderungen an die SOAP-Engine	20
3.4.2.	Bewertung der SOAP-Engines	20
4.	Apache Axis2	22
4.1.	Allgemeines	22
4.2.	Integration	23
4.3.	Zentrale Konzepte	24
4.4.	Handler und Module	26
4.4.1.	WS-Addressing	26
4.4.2.	WS-Security (Rampart)	26
4.5.	Bereitstellen eines Webservice	27
5.	Konzeption	28
5.1.	Analyse	28
5.1.1.	Schnittstelle SessionCube	29
5.1.2.	Das Axis2 Service-Archiv	29
5.1.3.	Services.xml	30
5.2.	Anforderungen	32
5.3.	Umsetzungskonzept	32
5.3.1.	Schnittstellenbeschreibung	33
5.3.1.1.	Mapping zwischen ActiveObject und Webservice-Objekt	34
5.3.1.2.	Validierung der Webservice-Objekte	35
5.3.2.	Verwaltung der Webservices	35
5.3.3.	Erzeugung eines Webservice	37
5.3.3.1.	Webservice-Objekt	38
5.3.3.2.	Webservice-Klasse	38
5.3.4.	Hilfsobjekte für Generierung und Mapping	39
6.	Implementierung	41
6.1.	Erstellung der Webservices im Backend	41
6.2.	Vaterklasse und Mapper Webservice<T>	43

6.2.1. Mapping Implementierung	44
6.2.2. Validierung der Objekte	46
7. Fazit und kritische Bewertung	47
Literaturverzeichnis	49
Eidesstattliche Erklärung	53
A. Anhang	i
A.1. WSDL Referenzen	ii
A.1.1. Liste weiterführender Literatur	ii
A.1.2. Aufbau eines WSDL 2.0 Dokument	ii
A.2. Quellcode Listings	iii
A.2.1. Aufbau einer WebserviceKlasse	iii
A.2.2. Aufbau eines WebserviceObjects	iv
A.3. Aufbau des Services.xml mit SSL-Unterstützung	v

Abbildungsverzeichnis

2.1. Architektur des GOF	8
2.2. Objekt-Attribut-Relation	10
3.1. Die Webservice-Architektur	14
3.2. Aufbau einer SOAP-Nachricht	15
3.3. Ablaufschema beim Code-First-Ansatz	18
3.4. Aufbauschema beim Contract-First-Ansatz	19
4.1. Axis2 Integration	23
4.2. Aufbau des Axis2-Verzeichnisses	24
4.3. Axis2 Architektur	25
5.1. Aufbau eines Service-Archivs	30
5.2. Erweiterte Architektur des GOF	34
5.3. Objektfluss im neuen Frontend (Mapping)	35
5.4. Aktivitätsdiagramm zur Webservice-Verwaltung	36
5.5. Aktivitätsdiagramm zur Generierung eines Service-Archivs	39
5.6. Aufbau der Frontend-Hilfsobjekte	40
6.1. Klassendiagramm der generischen Webservice-Vater-Klasse	43
A.1. Kompletter Aufbau eines WSDL2.0 Dokumentes	ii

Tabellenverzeichnis

3.1. Bewertung der verschiedenen SOAP-Engines	21
5.1. Beschreibung des Services.xml	31
5.2. Verwaltung der Webservices	37
5.3. Schritte zum Erzeugen eines Service-Archivs	38
6.1. Beschreibung der mapToT-Methode	45

Listingsverzeichnis

3.1. Aufbau eines vereinfachten WSDL 2.0 Dokument	16
5.1. Aufbau eines einfachen Services.xml	30
6.1. Erzeugung des WebserviceManager im Cube	41
6.2. Erzeugung des Service-Archivs	42
6.3. Aufruf der javac.exe mittels eines Prozesses	42
6.4. Methodenaufrufe der <i>Webservice<T></i> -Klasse	43
6.5. Konstruktoraufruf einer Webservice-Klasse	44
6.6. Mapping durch die <i>mapToT</i> -Methode	45
6.7. Mapping durch die <i>mapToActiveObject</i> -Methode	46
6.8. Aufruf der <i>Save</i> -Methode	46
A.1. Beispiel einer generierten WebserviceKlasse	iii
A.2. Beispiel eines generierten WebserviceObjektes	iv
A.3. Template des Services.xml mit SSL-Unterstützung	v

1. Einführung

Die vorliegende Bachelor-Thesis befasst sich mit der Erweiterung eines generischen Frameworks um Webservice-Fähigkeit. Das Generic Object Framework (kurz der GOF) ist ein hochkonfigurierbares System und die Software der sobedi informationsmanagement gmbh. Es soll ein neues Frontend erhalten, welches nicht nur wie bisher eine Mensch-Maschine-Kommunikation ermöglicht, sondern auch die Maschine-Maschine-Kommunikation.

In dieser Ausarbeitung werden verschiedene aktuelle Techniken und Verfahren für das zu entwickelnde Konzept betrachtet, diskutiert und bewertet. Den Abschluss bildet eine, auf dem Konzept entwickelte, Basisversion.

1.1. Motivation

Wiederverwendbarkeit, Kostenersparnis, hohes Abstraktionsniveau und leichte Anwendungsintegration sind wichtige Punkte bei der IT-Systementwicklung - eine Hilfestellung verspricht die Idee SOA.

Service-orientierte Architekturen (SOA) haben das Potenzial, das nächste bedeutende Paradigma der Informatik zu werden, und damit in die Fußstapfen der objekt-orientierten Programmierung zu treten. Die weiter abstrahierte Sichtweise, auf sonst sehr komplexe Systeme, ermöglicht diese Komplexität leichter zu bewältigen [Mel08, S.VII]. Durch die in den Vordergrund rückende Serviceorientiertheit können partnerübergreifende Integrationen von IT-Systemen leicht umgesetzt werden und eine hohe Wiederverwendbarkeit bei nicht redundanter Funktionalität erzielt werden. Flexible Geschäftsmodelle werden unterstützt und eine Automatisierung von Geschäftsprozessen ermöglicht [HA07].

Webservices, die Technik hinter der Idee SOA, stellen daher ein vielversprechendes Konzept für Unternehmen da, welches bereits von vielen großen Firmen¹ umgesetzt wurde.

1.2. Aufgabenstellung

Das Generic Object Framework ist ein hochkonfigurierbares System, womit schnell Software-Individuallösungen für Kunden erzeugt werden können. In einem für einen Endanwender aufgesetzten System handelt es sich um Businessobjekte (z.B. Auftrag, Person, Haus, uvm.), die in Beziehung zueinander stehen. Um diese Businessobjekte mit Applikationsdaten zu füllen, musste bisher auf eines von zwei bereits bestehenden Frontends zugegriffen werden. Derzeit gibt es eine webbasierte Benutzeroberfläche und eine Rich Client Plattform (RCP), die beide für die Mensch-Maschine-Kommunikation entwickelt wurden.

Im Rahmen dieser Bachelor-Thesis wird das vorhandene System um die Webservice-Fähigkeit erweitert und somit eine Maschine-Maschine-Kommunikation ermöglicht. Jedes Businessobjekt wird mittels einer Konfigurationseinstellung die Möglichkeit besitzen, via Webservice (siehe Kapitel 3.2) erreichbar zu sein. Die Webservices werden, wie die bereits vorhandenen Frontends, das Editieren, Löschen und Anlegen neuer Ausprägungen² ermöglichen.

Die über den Webservice übertragenen Objekte müssen dabei den Businessobjekten gleichen, d.h. der Name der übertragenen Objekte und deren Attribute müssen identisch sein. Eintreffende Objekte müssen, bevor sie an das bestehende System weitergeleitet werden, validiert werden. Die Benutzeridentifizierung, die bereits im vorhandenen System sehr komplex ist, muss ebenfalls Bestandteil der Erweiterung sein.

Eine detailliertere Beschreibung der Anforderungen wird im Kapitel 5.2 aufgeführt, da zu deren besserem Verständnis die nächsten Kapitel erforderlich sind.

¹Google, Amazon, eBay, uvm.

²ein Businessobjekt, das mit Applikationsdaten gefüllt ist, siehe Kapitel 2.3.2

1.3. Ziel der Arbeit

Generelles Ziel ist eine Erweiterung des bestehenden Systems um die Webservice-Fähigkeit mit Unterstützung eines Webservice-Toolkits.

Jedes der komplett unterschiedlich aufgebauten Businessobjekte kann seinen eigenen Webservice bereitstellen. Die Funktionalität eines Webservices besteht aus Auslesen, Erstellen, Editieren und Löschen von Ausprägungen, das Businessobjekt selbst kann dadurch nicht modifiziert werden.

Die Webservices können mittels einer Einstellung für alle Businessobjekte erstellt werden. Wird ein als Webservice verfügbares Businessobjekt im System überarbeitet, sollen diese Änderungen automatisch vom Webservice übernommen werden.

Die angestrebte Lösung befasst sich zunächst mit einfach aufgebauten Businessobjekten, ihrer Validierung und grundlegenden Tests. Komplexere Strukturen wie Tabellen, Charts, Diagramme oder Objekte im Objekt werden vorerst nicht berücksichtigt, sollen aber später leicht nachrüstbar sein.

1.4. Aufbau der Arbeit

Die Aufbaubeschreibung dient dem besseren Verständnis beim Lesen dieser Bachelor-Thesis und spiegelt im groben die Inhalte der einzelnen Kapitel wider. Dem Leser soll die Struktur und der Gedankengang dieser Ausarbeitung kenntlich gemacht werden.

Nach dem Einführungskapitel, das die Aufgabenstellung und das Umfeld beschreibt, enthält das zweite Kapitel eine Beschreibung des Generic Object Frameworks, das im Rahmen dieser Arbeit erweitert wird. Neben firmenspezifischen Begrifflichkeiten, werden Aufbau und (Arbeits)Umgebung erläutert.

Kapitel drei, Webservice-Grundlagen, geht zunächst auf Service-orientierte Architekturen und Protokolle von Webservices ein. Danach werden verschiedene Webservice-Toolkits (SOAP-Engines) betrachtet und bewertet. Das nächste Kapitel beschreibt die gewählte SOAP-Engine ausführlicher und erläutert dabei wichtige Details, die für den weiteren Verlauf notwendig sind.

Das Kapitel fünf umfasst die Analyse der Schnittstelle des vorhandenen Systems. Anschließend werden die genauen Anforderungen ermittelt, die für den späteren Soll-Ist-Vergleich dienen. Aufbauend auf den Anforderungen entsteht das Umsetzungskonzept, dessen Ergebnis die Grundlagen der zu entwickelnden Komponente bildet.

Den praktischen Teil der Implementierung stellt das sechste Kapitel dar. Hier werden nochmals Teile der zuvor im Konzept vorgestellten Lösungen aufgegriffen und Details hinsichtlich ihrer Implementierung aufgezeigt.

Abschließend erfolgt ein Fazit, das die entstandene Lösung kritisch betrachtet und diskutiert. Des Weiteren erfolgt ein Ausblick, welche Erweiterungen in künftigen Versionen implementiert werden könnten.

Die Fußnoten dieser Arbeit dienen als Hilfestellungen für ein leichteres Verständnis und führen zu Begriffserklärungen oder der offiziellen Webseite eines Produktes oder Organisation. Die angegebenen Webseiten dienen nicht, wie die im Literaturverzeichnis aufgeführten Werke, als Zitatquellen.

1.5. Firmenumfeld

Die sobedi informationsmanagement gmbh³ mit Sitz in Mannheim-Seckenheim wurde 1997 von dem Geschäftsführer Dirk Laufer gegründet und beschäftigt mittlerweile 37 Mitarbeiter. Ziel ist die Professionalisierung von Prozessen in der Projektabwicklung, im Wissensmanagement und im Beschaffungsmanagement. Hierfür werden komplette Leistungen von der organisatorischen Beratung und Konzeption, dem begleitenden Change-Management bis hin zur technischen Umsetzung geboten.

Neben der sobedi informationsmanagement gmbh gibt es noch die eng verzahnte Apinso GmbH⁴. In beiden Firmen geht es um die Entwicklung, Konfiguration und den Vertrieb des Generic Object Frameworks (siehe Kapitel 2.3).

Für die Erstellung oder Änderungen von Softwarelösungen gibt es Konfiguratoren, die im engen Kundenkontakt stehen. Das Entwicklungsteam programmiert in Reaktion auf neue Anforderungen weitere Funktionen, die nach erfolgreicher Testphase für die Konfiguratoren zur Verfügung stehen.

³<http://www.sobedi.com/> - Abruf: 12.12.2008

⁴<http://www.apinso.com/> - Abruf: 12.12.2008