

Aufgabe 02

Service Oriented Architecture

Patrick Mühl
Nanak Tattyrek

Inhaltsverzeichnis

Aufgabenstellung.....	3
Arbeitsteilung.....	4
Aufwandsschätzung.....	4
Zeitaufzeichnung.....	5
Git Log.....	5
Erfolge/Niederlagen.....	6
Technologiebeschreibung.....	6
Umsetzung und Testbericht.....	7
Quellen.....	7

Aufgabenstellung

Das neu eröffnete Unternehmen iKnow Systems ist spezialisiert auf Knowledgemanagement und bietet seinen Klienten die Möglichkeiten Daten und Informationen jeglicher Art (ähnlich wikipedia) in eine Wissensbasis einzupflegen und anschließend in der zentralen Wissensbasis nach Informationen zu suchen.

Folgendes ist im Rahmen der Aufgabenstellung verlangt:

- Stellen Sie ein Datenmodell für die Wissensbasis (so einfach wie möglich) auf.
[1 Punkte]
- Richten Sie 2 Services ein, implementieren und providen Sie diese:
 - sucheEintrag
in: suchwort
out: List
[3 Punkte]
 - erstelleEintrag
in: thema, content
out: success?
[3 Punkte]
- Implementieren Sie weiters einen Client, der Ihre Services verwendet. (Commandline/UI) [3 Punkte]
- Dokumentieren Sie im weiteren Verlauf den Datentransfer mit SOAP. [2 Punkte]
- Der Einsatz eines Enterprise Service Bus (eg. openESB) [4 Punkte]

Gruppengröße: 2

Abgabedokument:

Es sind alle allgemeinen Vorgaben zu erfüllen, sowie folgende spezielle Punkte

- Datenmodell
- Dokumentation der Services
- Dokumentation der Service-Clients

Anmerkung: (T.Micheler, 2013-10-25)

Bei der Abnahme DIESER Übung durch T.Micheler wird die Test-Coverage nicht berücksichtigt. Die Abnahme & der Test erfolgt im Unterricht!

Arbeitsteilung

Arbeitspaket	Patrick Mühl	Nanak Tattyrek
Wissensdaten bank		x
Search Funktion	x	
Insert Funktion		x
Protokoll	x	x
Client/Server	x	

Aufwandsschätzung

Arbeitspaket	Patrick Mühl	Nanak Tattyrek
Wissensdaten bank		0h30min
Search Funktion	1h	
Insert Funktion		1h
Protokoll	0h30min	0h30min
Client/Server	30min	
Gesamt:	2h	2h

Zeitaufzeichnung

Arbeitspaket	Patrick Mühl	Nanak Tattyrek
Wissensdaten bank		0h30min
Search Funktion	2h	
Insert Funktion		0h45min
Protokoll	0h30min	0h30min
Client/Server	0h20min	
Gesamt:	2h50min	1h45min

Git Log

```
commit ca5504e6d1b91a5050956cc727c4244cf53590dc
Author: Nanak Tattyrek <nanak@tattyrek.at>
Date: Thu Nov 21 22:28:47 2013 +0100

    Hinzufuegen von Daten implementiert

commit c5aebd0ba38cc9a842752f702367d8550cc5b332
Author: Patrick Muehl <pmuehl@student.tgm.ac.at>
Date: Thu Nov 21 21:03:31 2013 +0100

    Suchfunktion hinzugefügt

commit b0fc4d98f441961df8446b4425e431c5fc5618e1
Author: Nanak Tattyrek <nanak@tattyrek.at>
Date: Thu Nov 21 17:04:58 2013 +0100

    wissensdatenbank erstellt

commit c416aa9ec3b144c1634b1969a52fc77056b9a246
Author: Patrick Muehl <pmuehl@student.tgm.ac.at>
Date: Thu Nov 21 14:15:05 2013 +0100

    simple implementation of Client and Server

commit 2f8c0755acf62acee690f4d2cd919d2223532329
Author: Patrick Muehl <pmuehl@student.tgm.ac.at>
Date: Fri Oct 25 13:07:02 2013 +0200

    Added Package and started implementing

commit 9d66379197aad29a5e143f57e91164085da8d794
Author: Nanak Tattyrek <nanak@tattyrek.at>
Date: Fri Oct 25 02:48:24 2013 -0700

    Initial commit
```

Erfolge/Niederlagen

Erfolge: Die Implementierung des Clients und des Servers war recht schnell vollzogen.

Niederlagen: Das Lesen der XML-Datei dauerte etwas länger als gedacht. Dies entstand durch einen Denkfehler. Nach einer kurzen Erholungspause war der Fehler aber klar und konnte ausgebessert werden.

Technologiebeschreibung

„SOA ist eines der vielbeschworenen Buzzwords dieser Tage. Was dahinter steckt und welche pragmatischen Tipps ein Experte gibt, zeigt dieser Beitrag.

Die Idee hinter dem Konzept SOA, also den serviceorientierten Architekturen, klingt bestechend: die Programmlogik ist nicht mehr Bestandteil der einzelnen Programme, sondern wird über mehrere, unabhängige Services verteilt. IT-Prozesse werden so, vereinfacht dargestellt, als eigenständige Services definiert.

Prinzipiell kann man, ähnlich einer komfortablen Programmiersprache, auf einzelne fertige Module zurückgreifen und sich neue Anwendungen durch „Zusammenstecken“ erstellen. Dies funktioniert, weil alle Hersteller dasselbe Sechschichten-Modell verwenden. Eine pragmatische Standardisierung von Schnittstellen sorgt in Verbindung mit XML als einheitlichem Datenformat für die notwendige Interoperabilität.

SOA steht damit für die ganzheitliche Betrachtung einer IT-Systemlandschaft als Unterstützungsfunktion für betriebliche Prozesse. Funktionen, die durch einzelne Systeme abgedeckt werden, sind dank SOA in standardisierter Form unternehmensweit zugänglich. Einmal erstellte Abläufe oder Funktionen können, sofern sie die Schnittstellen aufweisen, also immer wieder verwendet werden. Aus ganzen IT-Prozessen werden so letztlich Programmierbausteine. Im Prinzip agieren die Dienste untereinander nach demselben Modell wie Webserver und Browser. Mit einer SOA können neue Systeme aufgebaut werden, die Teile der bestehenden IT nutzen. Aufgrund des unbestrittenen Nutzens dieser Standardarchitektur planen immer mehr Unternehmen die Einführung einer SOA. Besonders geeignet für Einführungsprojekte sind Vorhaben, die geschäftskritische Anwendungen betreffen, etwa Modernisierungs- und Migrationsprojekte, Datenintegrationsplattformen oder unternehmensweite, zentrale Lösungen wie beispielsweise Output-Management oder die Stammdatenverwaltung. SOA hilft so neben Zeit- und Kostenersparnis auch, die durchschnittliche Datenqualität in einem Unternehmen zu steigern.“ - [1]

Umsetzung und Testbericht

Server

Um den Server zu starten muss Python2 installiert sein. Des weiteren muss SOAPPy[2], eine externe Library von Python, vorhanden sein. Das durchsuchen des XML Files wird mithilfe von `xml.etree.ElementTree`[3] durchgeführt.

Der Server wird in der Konsole gestartet(`python SOAServer.py`).
Standartmässig läuft der Server auf localhost:8080. Man kann dies in der Quelldatei ändern.

Client

Der Client wird gleich wie der Server gestartet(`python SOAClient.py`).

Auch hier wird Python2 und SOAPPy vorausgesetzt. Der Client sendet Anfragen standartmässig auf localhost:8080.

Steht die Verbindung zum Server wird der User gefragt ob er Einträge lesen oder hinzufügen will

Wählt man 'hinzufügen' so kommen Abfragen zum Titel des Eintrags, dem Inhalt des Beitrags und den Tags die diesen Beitrag beschreiben.

Wählt man 'lesen', so wird der User gefragt welchen Beitrag er lesen möchte. Ist dieser in der XML- Datei vorhanden, so wird der Inhalt des Beitrages angezeigt. Ist der Beitrag nicht vorhanden, so wird dem User angezeigt das der gesuchte Inhalt nicht vorhanden ist

Quellen

[1]

http://www.tecchannel.de/webtechnik/soa/1725211/was_ist_eigentlich_eine_ser_viceorientierte_architektur_soa/index.html

[2]

<https://pypi.python.org/pypi/SOAPpy/>

[3]

<http://docs.python.org/2/library/xml.etree.elementtree.html>