

Logbuch von David Ferdinand Petersen

Web-basierte Anwendungen 2: verteilte Systeme

Montag 15.04.13

Am heutigen Tage hat der Workshop stattgefunden, welcher die Projektphase zwei einleitet. In der zweiten Phase geht es nun darum, sich sowohl mit dem zu verwendenden Datenformat, als auch das für die Übertragung notwendige Protokoll auseinander zu setzen. Die Aufgabe für heute bestand darin, ein Szenario zu formulieren, in dem sie synchrone aber auch asynchrone Datenübertragungsart Verwendung findet.

Entsprechend wurde versucht ein Szenario herauszuarbeiten, welches auch umsetzbar ist. Dazu wurden mehrere Vorschläge geliefert, welche teilweise verworfen aber auch übernommen wurden.

Zunächst begann ich zu überlegen, in wie fern solch eine Applikation aussehen könnte. Dazu habe ich mich bereits zu sehr in das System versetzt und bin so zunächst nicht auf ein konkretes Szenario gekommen, sondern habe Spezifikationen zum System ausgearbeitet. Nach dem Verwerfen dieser überlegte ich mir, unabhängig vom System und irgendwelchen Implementierungen, was für ein Szenario man auf synchroner und asynchroner Ebene verwenden kann. Unter Anderem hatte ich die Idee eine Applikation bezüglich Bus-Linien zu erstellen, in denen der Nutzer Abfahrtszeiten, Planänderungen, etc. mitgeteilt bekommt. Aufgrund mangelnder Dynamik (Planänderung alle halbe Jahre) wurde sich schließlich innerhalb der Gruppe darauf geeinigt, das Szenario mit den Sportveranstaltungen zu wählen.

Bei den Sportveranstaltungen herrscht meiner Meinung nach immer Planänderungen oder Ausfälle von Trainingseinheiten etc.. Aus diesem Grund bat es sich an, ein Szenario wie dieses zu Verwenden, wo der Benutzer auch zeitnah über Änderungen seiner abonnierten Veranstaltungen informiert wird.

Donnerstag 18.04.2013

Heute habe ich mich mit meinem Gruppenmitglied getroffen, und wir haben gemeinsam das Szenario ausgearbeitet, bzw. genauer betrachtet und ein System als Lösung für dieses entwickelt. Daraufhin haben wir bereits nebenläufig Ideen für mögliche Ressourcen gesammelt. Ich habe ein Kommunikationsdiagramm erstellt, indem ersichtlich ist, wie ein Nutzer mit dem System interagieren kann bzw. welche Anfragen wann an den Server gesendet werden können. Laura hat den Text zu dem Kommunikationsdiagramm verfasst.

Dienstag 23.04.2013

Heute habe ich mich wieder mit meinem Gruppenmitglied getroffen. Wir haben angefangen Ressourcen auszuarbeiten, und die dazugehörigen Schritte aus der Vorlesung beachtet. Daraufhin haben wir später noch das XML-Schema angefangen zu schreiben, welches Laura dann im Laufe fertigstellen wird.

Samstag 27.04.13

Heute habe ich Anpassungen am Schema vorgenommen (ausführlicher: siehe Git). Des Weiteren habe ich mir überlegt: Eigentlich sollte der Benutzer keinen eigenen Account bekommen, wie es z.B. ein Veranstalter tun sollte, jedoch muss der Benutzer ja ebenfalls authentifiziert werden können, falls er sich beispielsweise als Veranstalter eintragen möchte, andere Veranstaltungen, Veranstalter etc. abonniert,...

Wir müssen also nochmals klären, ob wir nun den Benutzer vom Veranstalter unterscheiden, mit Attribut? Als eigenes Element? ...

Montag 29.04.13

Heute haben wir uns bei dem Praktikum getroffen um Fragen wegen der Realisierung der Verknüpfungen der Ressourcen in XML zu klären. Des Weiteren wurden allgemeine Fragen geklärt. Ausserhalb des Praktikums haben wir in der Gruppe im MI-Pool zusammen weitergearbeitet und die Ressourcen überarbeitet, und das Schema angepasst.

Mittwoch 01.05.13

Heute habe ich den Dokumentationsteil 'Ressourcen' angefangen zu schreiben.

Sonntag 05.05.13

Heute habe ich den Teil der Dokumentation, welche die Ressourcen (Verben ...) betrifft nahezu fertig gestellt.

Montag 06.05.13

Heute haben wir nach der Vorlesung im Praktikum weiter an unserem System gearbeitet und mit den Mentoren unser Schema sowie unsere Ressourcen und deren Operationen (Verben) diskutiert. Als nächstes Ziel ist die Entwicklung einer Webapplikation.

Dienstag 07.05.13

Heute haben wir uns in dem Team getroffen um den Webserver zu erstellen. Dabei hat Laura sich um das unmarshalling und alles zugehörige gekümmert und ich habe mich an den Webserver mit Grizzly versucht. Dabei fiel uns gestern ein, dass das Schema noch nicht ganz sauber war. Daraufhin haben wir es nochmals angepasst. Mit dem jetzigen Schema ist dieses zwar an unseren Vorstellungen angepasst, jedoch kann ich mir vorstellen, dass es noch Schwierigkeiten mit sich bringt, da eine Veranstaltung über die zugehörige Sportart und diese über die zugehörige Sportgruppe definiert wird. Somit muss man sich durch diese Elemente „durchkämpfen“ bis man bei der Veranstaltung landet. Dies könnte eventuell dann Schwierigkeiten mit sich bringen, wenn man einen Veranstalter kennt und zu diesem seine ausstragenden Veranstaltungen anzeigen zu lassen. Denn dann müsste man alle Veranstaltungen durch alle Sportarten und -gruppen durchforsten und dann die ID der Veranstaltungen mit dem im Element Veranstalter zugehörigen Referenzierungselement vergleichen ... es wird spannend ..

Mittwoch 08.05.13:

Nach Diskussion und Ausarbeitung des neu überdachten Schemas wurde dies nun im Projekt eingefügt und wird verwendet. In der Dokumentation finden sich Vor- und Nachteile dieser Herangehensweise. Heute habe ich weiter an dem Webserver versucht zu bauen, habe die Dokumentationen gelesen, habe jedoch keinen Erfolg verspüren können.

Donnerstag 09.05.13

Auch heute habe ich wieder mich an den Webservice versucht. GET-Abfragen, welche Strings zurückliefern sind nun endlich implementiert. Als nächstes habe ich mich an PUT-Anfragen versucht. Leider bisher erfolglos.

Freitag 10.05.13

Webservice....rauf und runter. Da PUT immer wieder scheiterte, fing ich an die DELETE-Operation zu implementieren. Und danach die POST Methode anzufangen.

Samstag 11.05.13

POST fertiggestellt, danach die restliche Zeit mit PUT und Optimierungsarbeiten verbracht.

Sonntag 12.05.13

Mit Laura Doku geschrieben, nebenbei einige Anpassungen am Programm vorgenommen (für Details siehe Git).

Montag 13.05.13

Wir haben sehr gutes Feedback bekommen. „Sehr vorbildlich und fleißig“ waren wir. Bis auf den 3. Meilenstein war bis dahin auch unsere Dokumentation sehr fortgeschritten. Nach der „Abnahme“ haben wir zusammen überlegt und recherchiert, wie der nächste Meilenstein mit XMPP-Vorbereitungen ablaufen könnte.

Sonntag 19.05.13

Heute habe ich den XMPP Server installiert und die entsprechend benötigten Libraries zum Java Projekt hinzugefügt. Leider ist mir noch unklar, in wie fern der bisher erstellte (REST) Server mit dem neuen XMPP Server harmonisieren soll. Werden diese unabhängig voneinander implementiert, sodass der spätere Client dann Verbindungen zu zwei Servern aufbaut? Oder greift der XMPP Server auf die Daten des REST-Servers zu? ...

Dienstag 28.05.13

Heute habe ich mit Laura versucht am XMPP Client zu arbeiten. Dazu haben wir Literatur herangezogen die uns bisher leider noch nicht weitergebracht haben. Dafür haben wir an der Doku Fortschritte erzielen können und der RESTful Webservice wurde nochmal um einige Erweiterungen und Fehlerverbesserungen angepasst.

Mittwoch 29.05.13

Auch heute haben wir wieder am Client für den XMPP-Server gearbeitet. Dabei habe ich entschlossen zunächst einmal lediglich die Methoden vorzufertigen, sodass später dann diese von der eigentlichen Client-Applikation eingebunden werden kann.

Donnerstag 30.05.13

Da wir leider immer noch Probleme mit dem Klienten haben, insbesondere, fragen wir uns noch, in wie fern der Payload verwendet wird/werden soll. Unabhängig davon haben wir jedoch zunächst die Methoden weitergeschrieben, sodass wir morgen dann als Ziel gesetzt haben, den Klienten zwar fertig zu stellen, jedoch ob dieser sich auf den bisher erstellten REST Server beziehen kann ist noch unklar.

Freitag 31.05.13

Wie erwartet, wurde der Client immerhin fertig gestellt, jedoch besteht weiterhin das Problem, dass der XMPP-Server und Client zunächst unabhängig von dem bisher erstellten REST-Servers abläuft. Das heißt, es sind alle vom Meilenstein 4 geforderten Funktionen implementiert, jedoch sind diese zunächst auf Konsolenbasis realisiert und beziehen sich nicht auf bereits bestehende Ressourcen.

Freitag 07.06.13

Jetzt, da der XMPP-Server abgearbeitet worden ist, ist der GUI Client dran. Diese Aufgabe wird weitestgehend von Laura übernommen. Ich kümmere mich derweil darum, dass die Methoden an den 'echten' Client angepasst werden. Das heißt, die vorher definierten REST Methoden (GET, POST,...) werden so angepasst, dass diese nun auch von dem von uns erstellten Client verarbeitet werden können. Hierbei ist die Schwierigkeit, dass wir bei einer Listenressource eine (Array)Liste per GET übergeben, und nicht wie vorher, der Einfachheit halber einen konkatenierten Sting. Dies bringt momentan noch massive Schwierigkeiten mit sich.

Mittwoch 12.06.13

Nach dem Gespräch mit den WBA-Mentoren bezüglich des angesprochenen Problems hinsichtlich des Übergebenen Strings (welcher schon Unmarshallt wurde), wo jedoch zunächst reiner (unmarshallter) XML-Code übergeben werden sollte, stellte sich heraus, dass dies nun im Programm nochmal komplett überarbeitet werden soll.

Feedback für die Veranstaltung: Den Client parallel zu den Web-Funktionen schreiben, sodass nicht ganz am Ende Probleme auftauchen, die eigentlich schon viel früher (im REST-Meilenstein) auftreten sollten.

Im Laufe der Tage bis hin zur Abgabe wird also nochmal der gesamte REST-Service angepasst... hoffentlich passiert das bei dem XMPP-Server nicht auch, jedoch hatte ich bei dem XMPP-Server mehr Bedenken als beim REST-Server ...

Donnerstag 13.06.13

Heute habe ich angefangen den REST-Service umzucoden, sodass dieser nun für den späteren Client eine ordentliche Funktion wiedergibt. Damit habe ich mich den Tag über beschäftigt und habe vor das Umschreiben morgen fertigzustellen.

Freitag 14.06.13

Ich habe es geschafft die gesamten REST-Methoden umzucoden. Dabei wurden die zunächst sowieso XML verarbeitenden Methoden nicht umgeschrieben, da diese zum Glück ohne weiteres funktionieren. Der Client, welcher die neu geschriebenen Methoden nutzt, ist noch in Arbeit. Über das Wochenende soll der Client komplett fertig sein, am besten in der GUI bereits integriert. Am besten ist dann auch schon geklärt, ob der bisher bestehende XMPP-Server ebenfalls noch angepasst werden muss. Da dies höchstwahrscheinlich der Fall ist, sollten nach Möglichkeit bis Montag alle Fragen dazu geklärt werden, sodass im Beratungstermin dann nochmal wahrgenommen werden kann, damit bis Freitag (Sonntag?) dann auch alles fertig ist.

Samstag 15.06.13

Es wird unter Hochtüren am System gecoded. Da die GUI noch nicht ganz fertig ist, habe ich schonmal Methoden geschrieben, auf die der Client dann einfach zurückgreifen kann. Ich habe ein kleines Script geschrieben, welches es ermöglicht, ein komplett neu aufgesetztes System, alle benötigten Leafs hinzufügen zu lassen. Der REST Client ist nun also fertig. Die Methoden für das Löschen, Hinzufügen und Aktualisieren wurden hinzugefügt. Der XMPP-Service wurde auch schon leicht an die spätere GUI angepasst, indem die Methoden so abgeändert wurden, dass sie in der GUI verwendet werden können. Des Weiteren wurden weitere nützliche Methoden für den XMPP-Service hinzugefügt, sowie der Server um Test-Nodes bereinigt, sodass nun nur noch LeafNodes und Subscriptions hinzugefügt werden können, die im späterem Gebrauch auch sinnvoll und nutzbar sind. Morgen wird dann der Rest für XMPP geschrieben. Ich hoffe dann ist die GUI fertig und wir können spätestens Montag durchstarten, das Programm fertig zu schreiben, also die Funktionalität mit der GUI mergen.

Sonntag 16.06.13

Für die Rest-Service-Klassen habe ich heute nochmal die JavaDoc angepasst, sodass jede Methode und jede Klasse Informationen aufweisen, welche für das spätere Implementieren in die GUI nützlich sind, und einen großen Teil der Programmdokumentation ausmachen.

Leider habe ich es nicht geschafft den XMPP-Service fertig zu stellen, da die Items, welche beim Publishen übertragen werden, nicht die eigentliche Message enthalten, bzw. nicht auf den Content zugegriffen werden kann.

Montag 17.06.13

Nach der WBA Vorlesung ging es dann zum Beratungstermin:

Wir fragten nach dem Payload, welcher nicht mehr aus ein gepublishtem Item ausgelesen werden konnte, sowie nach dem Bug, dass max ein Item gepublisht werden kann und alle anderen nicht weiter beachtet werden, bzw. dass der Listener nicht reagiert. Leider konnten wir keine Lösung zu diesen Problemen finden es stellte sich lediglich heraus, dass die Smack-API voller Bugs ist. Toll .. nun wie auch immer, den Rest des Tages habe ich mich damit beschäftigt, es zu versuchen den Payload irgendwie zu übertragen.

Der Payload wird nun über das RootElement des Items übertragen. Das ist zwar nicht Sinn der Sache, jedoch komme ich so einfach über die Methode '`getElementName()`' an den gewünschten Inhalt heran.

Nun bleibt trotzdem noch das Problem mit dem Item-Publishen.

Dienstag 18.06.13

Die Zeit bleibt einfach nicht, Lösungen für halbfertige API's zu umschreiben. Es geht einfach nicht mehr als ein Item zu publishen. Der gesamte Tag ging mal wieder nur für Problemfindung und Rumprobiererei hinüber. Schade, dass der Support hier nicht ausreichend ist, wenn wir schon so eine halbfertige API verwenden :(

Immerhin haben wir es geschafft, einen kleinen Teil der Services in der GUI zu implementieren.

Mittwoch 19.06.13

Heute habe ich etwas an der Doku hinsichtlich der Anfertigung des Programmes geschrieben, die JavaDoc für die Webservices weitestgehend fertiggestellt und die GUI für den Interessenten nahezu mit allen Funktionalitäten ausgestattet.

Donnerstag 20.06.13

Bei weiterer Implementierung der Services in die GUI treten immer mehr Fehler auf. So kommen teilweise lauter Exceptions, bei denen einige nicht auffindbar gemacht werden können. Es stellte sich heraus, dass die ganzen ActionListener der GUI am besten direkt oben im Programmcode eingefügt werden sollten, da die häufige Veränderung des GUI-Panels doch zu verzwickten Exceptions führen kann. Nachdem immer wieder einige Probleme gelöst worden sind, sind wir immerhin so weit gekommen, dass der Interessent bereits über Publishings Meldungen in der Standardausgabe Bescheid bekommt. Morgen sollte das Programm dann endgültig fertig gestellt werden, auch, wenn einige Fehler dennoch erscheinen können.

Freitag 21.06.13

Wie gewünscht, haben wir heute das Programm fertig gestellt. Leider bestehen immernoch einige Fehler, wie, dass ein Dropdown-Feld bereits eine vordefinierte Auswahl enthält, welche trotzdem noch angewählt werden muss, damit eine Funktion ausgeführt wird. Des Weiteren entsteht immer eine Exception, wenn man im Programmcode zu schnell klickt. Ich vermute mal, dass das daran liegt, dass wir immer wieder den ActionListener vom Dropdown-Feld entfernen und hinzufügen; je nachdem ob das Dropdown-Feld benutzt werden kann oder nicht. Wenn nun schnelle Klicks zu vielen Aktionen führen, kann ich mir vorstellen, dass das Dropdown-Feld mehrere ActionListener zugeteilt bekommt, und so das Programm irgendwie eine Exception wirft. Glücklicherweise ist dies, bis auf die geworfene Fehlermeldung, nicht weiter tragisch; das Programm läuft trotzdem weiter

und funktioniert auch.

Samstag: 22.06.13

Nun werden nur noch Feinarbeiten am System vorgenommen; es wurde ersichtlich, dass beim Starten des Programms nach vorherigem Subscriben, die Nachricht neu gefeuert wird und somit die Subscriptions-Liste falsch dargestellt wird. Des Weiteren könnte man noch den Fehler mit dem Dropdown-Feld beheben, und anstelle von ActionListenern, könnten besser Item-Listener verwendet werden. Vielleicht würde sich dann das Problem mit den Exceptions beheben, da dann der ActionListener nicht immer wieder hinzugefügt und entfernt werden muss. Dennoch wird sich zunächst primär auf die Dokumentation konzentriert.

Sonntag: 23.06.13

Die Fehler von gestern wurden behoben, jedoch treten ab und zu immernoch Exceptions auf, die das Programm jedoch nicht weiter bei der Ausführung stören. Schließlich wurde die Dokumentation heute fertig gestellt und damit das Projekt abgeschlossen.