Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering

Integration eines Experience Forums in das SOA-Me-System

Studienarbeit

im Studiengang Mathematik mit Studienrichtung Informatik

von

Sabina Grimm

Prüfer: Prof. Dr. Kurt Schneider
Betreuer: Dipl.-Wirt.-Inform. Daniel Lübke

Hannover, 10. April 2007

Danksagung

Ich bedanke mich bei Herrn Daniel Lübke für die hervorragende Betreuung und Unterstützung meiner Studienarbeit.

Inhaltsverzeichnis

1 Einleitung 2

1.1 Serviceorientierte Architektur 2
1.2 Erläuterung 2
1.3 Ziel dieser Arbeit 3
1.4 Gliederung dieser Arbeit 3

2 SOA-Me-System 4

3 Experience Forum 6

3.1 Beschreibung 6
3.2 Einbindung von Experience Forum in die Softwareentwicklung 8
3.3 Funktionen und Layout von Experience Forum 10
3.3.1 Feedbacktypen 10
3.3.2 Bewerten von Einträgen 13
3.4 Aussehen des Experience Forums 14

4 Implementierung 15

4.1 Beschreibung der Architektur 15
4.2 Anforderungen 16
4.2.1 Funktionale Anforderungen 16
4.2.2 Qualitätsanforderungen 17
4.3 Feedback-Datenbank 17
4.4 Beschreibung der Webservices 18
4.5 Verwaltung im Experience Forum 20

5. Anwendungsszenario 23

6. Zusammenfassung und Ausblick 25

Literaturverzeichnis 26

Abbildungsverzeichnis 27

Anhang A: Use Case Beschreibungen 28

Anhang B: Webservice Definition 30

Anhang C: Installation und Konfiguration 35

1 Einleitung

1.1 Serviceorientierte Architektur

Serviceorientierte Architektur (SOA) ist eine Architektur für Anwendungssysteme zur durchgängigen Unterstützung von Geschäftsprozessen. Die Software sieht dafür lose gekoppelte Dienste vor, die so genannten Services. Jeder Service bietet eine Funktionalität in dem Geschäftssinn an, welche sich direkt auf eine Geschäftsgelegenheit oder -funktion bezieht. Diese Services sind zu so genannten Kompositionen in laufenden Prozessen gruppiert. Die am weitesten bekannte Technologie um SOA zu realisieren sind Webservices.

1.2 Erläuterung

Die Entwicklung von Geschäftssoftware erfordert meistens eine Verbesserung in mehreren Zyklen. Es sind viele Anwender und Stakeholder einbezogen, wie auch eine Vielzahl von umfangreichen und komplexen Geschäftsprozessen. Eine schrittweise Verbesserung kostet Zeit und Geld, wenn die Probleme nur langsam erkannt werden. Ein Ansatz ist es, ein Experience Forum in das System einzubinden. Durch feste Einbindung des Erfahrungsaustauschs profitiert man von der nahen Beziehung zu den Geschäftsprozessen. Gleichzeitiges Generieren von Benutzeroberfläche und dem Forum bietet Gelegenheiten zum automatischen Einteilen und Zuordnen von Rückmeldungen. Verbesserte Rückmeldungen und eine effektive Kommunikation könnten die Entwicklung von umfangreichen SOA-basierten Softwaresystemen beschleunigen und zudem im Preis senken.

Informationssysteme wurden entwickelt um eine große Anzahl von Benutzern in prozessabhängigen Anwendungen zu unterstützen. Deshalb müssen diese flexibel gestaltet werden und brauchen eine erhebliche Anpassung an die Geschäftsprozesse. Sie müssen nicht nur die Prozesse ausführen, sondern auch die Anwender beim Ausführen ihrer Aufgaben innerhalb eines Prozesses unterstützen. Das bedeutet, dass der Benutzer durch den Prozess geführt werden soll und ihm bei den nicht alltäglichen Aufgaben geholfen wird. Verschiedene Faktoren, wie Aussehen, Flexibilität und Funktionalität von den Geschäftsprozessen und Onlinehilfe sind bei dieser Betrachtung wichtig.

Korrekte Gestaltung solcher Elemente ist aufgrund von umfangreicher und ungleicher Anwenderbasis und der Formkomplexität der Geschäftsprozesse problematisch. Normalerweise ist der formale Prozess anders als der gewohnte. Deshalb wird Kommunikation zu einem Erfolgsfaktor, besonders unter den oben aufgeführten Umständen. Das ist der einzige Weg wie Software und

geschäftsbezogene Probleme gefunden und in den Entwicklungsaufbau zurückgekoppelt werden können. Diese werden dann beim weiteren Schritt behoben. Der Erfolg und die Anerkennung des endgültigen Produkts hängen von der Zusammenarbeit mit den Anwendern ab.

1.3 Ziel dieser Arbeit

Ziel dieser Arbeit ist es ein solches Experience Forum zu entwickeln, welches in den bereits vorhandenen Client der SOA-Me-Plattform integriert werden soll. Dabei werden die Erfahrungen über eine Webservice-Schnittstelle in einer Datenbank verwaltet. Zusätzlich sollten solche Aspekte wie Platzmangel und Übersichtlichkeit bewältigt werden.

1.4 Gliederung dieser Arbeit

Die nachfolgenden Abschnitte sind folgendermaßen gegliedert:

Abschnitt 2 beschreibt das SOA-Me-System. Es wird auf die bereits vorhandenen Grundlagen eingegangen.

Abschnitt 3 befasst sich mit dem Experience Forum. Hier wird die Einbindung vom Forum in die Softwareentwicklung, sowie seine Funktionen beschrieben.

In **Abschnitt 4** folgt eine Beschreibung der Implementierung.

Anschließend wird in **Abschnitt 5** ein Anwendungsbeispiel und in **Abschnitt 6** eine Zusammenfassung und Ausblick gegeben.

2 SOA-Me-System

Im Rahmen eines Software-Projekts wurde ein SOA-Me-System entwickelt, das aus drei Komponenten – dem SOA-Me Client, dem SOA-Me Integrationsserver und dem SOA-Me Kompositionseditor – besteht (s. Abb. 1). Ziel dieses Projektes war ein System, das beliebige automatisierbare und vor allem nicht automatisierbare Geschäftsprozesse unterstützt, die von dem Kompositionseditor unkompliziert definiert werden können. Ausgeführt werden diese Prozesse über den Integrationsserver, der dabei auf Webservices und User-Interaktionen, die über den Client ermöglicht werden, zugreift. Alle drei Teilsysteme arbeiten in einem Netzwerk. Dabei kann es sich sowohl um das Internet als auch ein Intranet handeln.

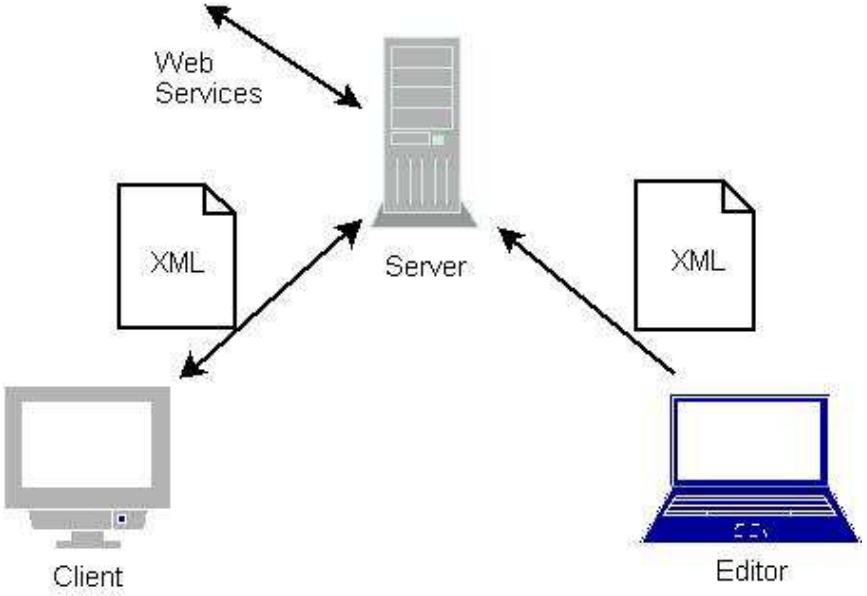

Abbildung 1: SOA-Me-System Architektur

Die Geschäftsprozesse basieren auf Ereignisgesteuerten Prozessketten (EPK) und werden im Kompositionseditor modelliert. Anschließend werden sie an den Server exportiert, auf dem diese Prozesse gemäß der Semantik von EPKs ausgeführt werden.

Der Server seinerseits kümmert sich um die Benutzerverwaltung. Zusätzlich ist er für die Webservice-Aufrufe, die die Benutzerinteraktionen steuern und verwalten, zuständig. Außerdem sind die Listen von Prozessen und Todo-Aufgaben auf dem Server abgelegt. Für jeden Benutzer variieren die Prozesslisten in Abhängigkeit davon, ob er die Prozesse starten darf. Die Todo-Liste enthält die Aufgaben zu den bereits gestarteten Prozessen. Eine weitere Funktion des Servers ist die Dokumentation des Ablaufes eines Prozesses.

Als Benutzerschnittstelle dient der Client (Abb. 2). Er bietet dem Anwender eine benutzerfreundliche Oberfläche und visualisiert graphisch die Abläufe auf dem Server. Mit dem Client kann sich ein User am Server anmelden, Geschäftsprozesse starten und daran teilnehmen. Zu jedem gestarteten Prozess arbeitet der Benutzer eine Reihe von Aufgaben ab. Dabei wird das User Interface abhängig von der gewählten Aufgabe aus einer vom Server geschickten XML-Datei dynamisch generiert.

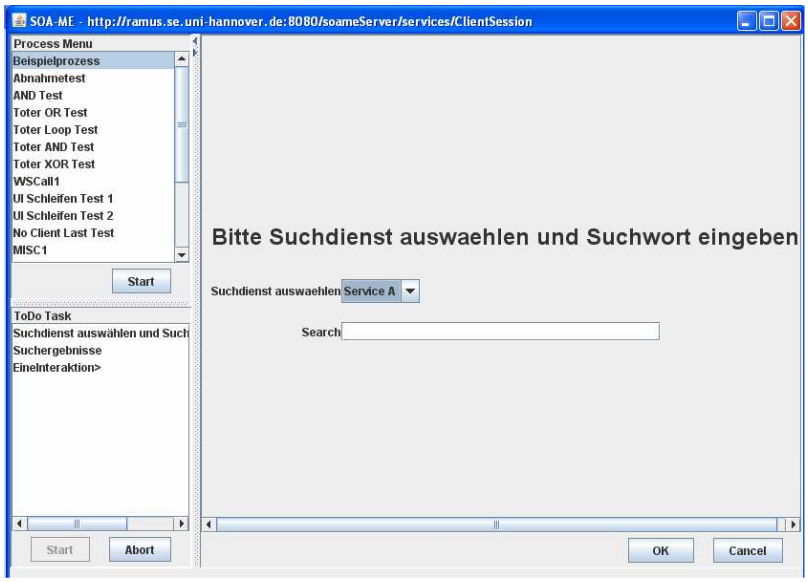

Abbildung 2: SOA-Me Client

3 Experience Forum

3.1 Beschreibung

Während der Softwareentwicklung passieren oft Fehler bei der Anforderungsanalyse. Dabei werden Anforderungen falsch interpretiert oder umgesetzt. Außerdem treten auch nicht selten technische Probleme bei der Benutzung der Software auf, das man auf die Codierung zurückführen kann. Bei der neuen Version werden schließlich diese Probleme bestmöglich behoben. Es entsteht ein iterativer Prozess aus Entwicklung und Anwendung. Um die Fehler erfassen zu können, benötigen die Entwickler Feedback von den Anwendern. Die Erfahrungen werden ab dem ersten Anlauf gesammelt. Das Experience Forum sammelt solche Erfahrungen und bietet den Entwicklern Grundlagen zu Verbesserungsansätzen. Die Grundidee ist, die Anwender in den Entwicklungsprozess miteinzubeziehen. Die aus dem Experience Forum entsprungenen Verbesserungen können dann ab bei der zweiten Iteration hinzugefügt werden.

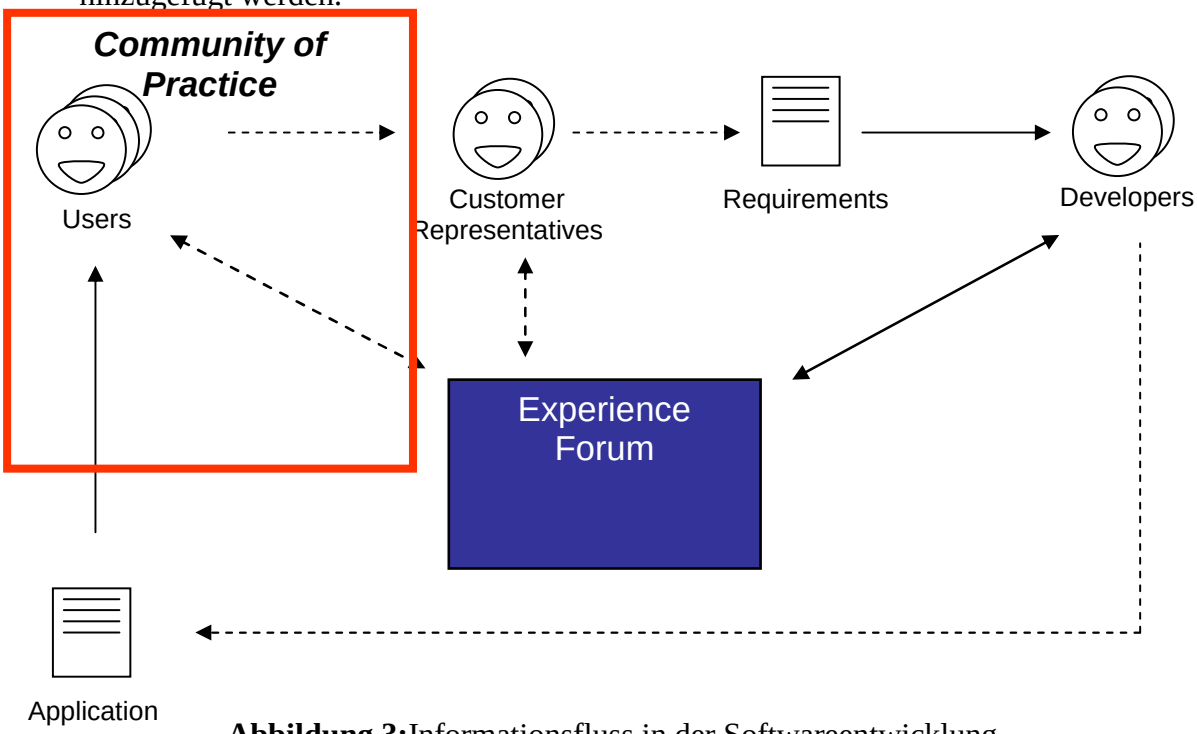

Abbildung 3: Informationsfluss in der Softwareentwicklung

Wie die Abbildung 3 zeigt, erreicht das Feedback der Anwender die Entwickler ohne das Experience Forum erst nach drei informationsproblematischen Instanzen [1]. Beim ersten Fluss müssen Personen die Initiative ergreifen und sich

mit der Kommunikation befassen. Die Stakeholder müssen die geänderten Anforderungen niederschreiben. Nach einiger Zeit brauchen die Entwickler mehr Aufwand um die geänderten Anforderungen zu lesen und zu verstehen. Die Anforderungen gehen durch drei Instanzen: Merken, Schreiben und Lesen. Falls eins der drei misslingt oder unterlassen wird, erreichen die Anforderungen nicht das Entwicklungsteam. Aber selbst wenn es das Entwicklungsteam erreicht, werden die umgewandelten Anforderungen oft missverstanden. Die Umwandlung findet mit einem Personen-, Zeitaufwand und unklaren Aussichten statt.

Es gibt verschiedene Lösungsansätze für dieses Problem:

- Der Kunde vor Ort überbrückt die Verbindung zwischen der Anwendergemeinschaft und den Entwicklern. Dies funktioniert aber nur dann, wenn der Kunde die Gemeinschaft gut genug vertreten kann. Die SOA-Anwendungsysteme werden von einer breiten und heterogenen Benutzergruppe benutzt. Zusätzlich kann der Kunde nicht alle Aspekte wie Benutzerfreundlichkeit und Funktionalität gleichzeitig berücksichtigen. Eine einzelne Person kann nicht eine breite Gruppe vertreten.
- Anwendungssysteme sollen den Benutzern die Gelegenheit geben in einfacher Art ihre Erfahrungen und Rückmeldungen mitzuteilen. Sie müssen wissen, an wen sie ihre Beiträge adressieren und schnelle Antworten z.B. in Form von Verbesserungen erhalten.

Die Geschäftsprozesse in SOA-Me-Systemen werden wie bereits beschrieben von dem Designer modelliert, während die entsprechende Software von einem Team aus Entwicklern geschrieben wird. Als Endverbraucher steht der Anwender dem gesamten System gegenüber. Dabei passiert es oft, dass Fragen wie „Wie nutze ich diese Applikation richtig?“ oder „Wie bearbeite ich diese Aufgabe am besten?“ auftreten. Beim ersten Ansatz besteht keine direkte Verbindung zwischen diesen drei Gruppen, was aber notwendig ist, um ein bestmögliches Resultat für den Benutzer zu erzielen. Beim Schreiben der Software kann der Entwickler nicht auf die Wünsche und Anforderungen der Anwendergemeinschaft eingehen, sondern berücksichtigt dabei die des Kunden. So entsteht dabei eine nicht so funktionale und benutzerfreundliche Applikation wie erhofft. Das gleiche Phänomen tritt auch beim Designen der Geschäftprozesse auf.

Beim zweiten Lösungsansatz muss die Kommunikationsverbindung zwischen den Endverbrauchern und dem Entwicklerteam verbessert und hergestellt werden. Weil die Information an mehreren Stellen durch die beteiligten Personen zerstreut wird, sollte die Kommunikation technisch umgesetzt werden. Es ist ferner notwendig den Aufwand zum Erstellen von Feedback für die Verbesserung von Software und der Geschäftsprozesse zu reduzieren.

Eine Lösung für dieses Problem ist das Experience Forum: Es erhöht den Kommunikationsfluss zwischen den drei Gruppen und bietet eine gemeinsame Basis. Durch das Forum können die Benutzer ihre Anregungen und Wünsche direkt den Entwicklern bzw. Prozessdesignern mitteilen. Zusätzlich bietet sich den

Anwendern die Möglichkeit Erfahrungen, z.B. zur besseren Handhabung der Geschäftsaufgaben, untereinander auszutauschen. Im Hintergrund stehen die Entwickler und die Designer, die beim nächsten Anlauf Rücksicht auf die Wünsche und Vorschläge der Anwender nehmen können.

Die neuen Anforderungen entstehen aus dem Umwandeln und Verfeinern von Feedback. Nächste Aufgabe ist das Sortieren und Priorisieren der neuen Anforderungen und das Weiterleiten an den Prozessdesigner oder an die Softwareentwickler. Die Prozessdesigner passen den Geschäftsprozess an, was normalerweise zusätzliche Änderungen an der Software erzeugt.

Um die Benutzung des Experience Forums zu erleichtern und zu forcieren ist es wichtig, dass es direkt in den Client integriert wird. In diesem Fall soll es in den SOA-Me-Client als Seitenleiste integriert werden. Auch erlaubt das Experience Forum das Einfügen neuer Erfahrungen, Vorschläge und das Hinweisen auf Programmfehler. Außerdem können die Anwender die Beiträge anderer Nutzer bewerten. Das kann zum Sortieren genutzt werden. Das ist wichtig, da der Platz für das Forum beim Client begrenzt ist. Dabei werden die besten und interessantesten Erfahrungen zuerst angezeigt. In vielen Foren muss der Anwender seine Beiträge zuordnen. Beim Experience Forum geschieht dies automatisch, denn die Erfahrung wird automatisch einer Prozessaufgabe zugewiesen. Die komplette Information ist mit der aktuellen Aufgabe verbunden. Das bedeutet, dass neben dem dynamischen Generieren von User Interface auch in Abhängigkeit von der Aufgabe das Forum neu gestaltet wird.

3.2 Einbindung von Experience Forum in die Softwareentwicklung

Entwickler brauchen oft nicht nur technische Fertigkeiten, sondern sollten auch über das Wissen des Einsatzgebiets und Benutzeroberflächentauglichkeit verfügen. Wann immer viele Interessenten einbezogen sind, gelingt es selten alle Anforderungen des Kunden beim ersten Versuch zu treffen. Es dauert meistens lange ein System zu bilden, anzuwenden und wieder zu verbessern.

Während das Experience Forum Benutzerrückmeldungen sammelt, kann es in die Entwicklung eingebunden werden, um das Projekt zum Erfolg zu führen [2]. Das Forum kann nach der ersten Version bei den Anwendern genutzt werden, oder aber auch bei der vorläufigen Version bei einer begrenzten Anzahl von Nutzern.

Das Experience Forum kann in jede Phase der Anforderungsanalyse (Abb. 4) eingebunden werden:

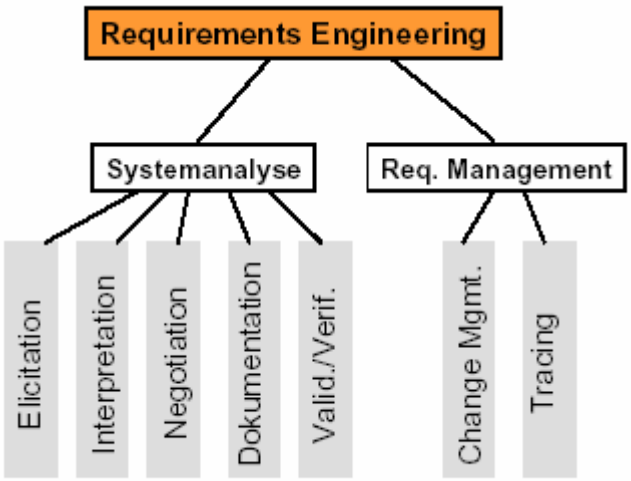

Abbildung 4: Phasen der Anforderungsanalyse

Anforderungsanalyse :

- Elicitation (Herausfinden):

Die Benutzer werden ermuntert ihre Meinung zu schreiben. Der Entwickler muss letztendlich die Anforderungen sortieren: Es gibt Anforderungen, die die Software und diejenigen die das Geschäftmodel betreffen.

- Interpretation:

In dieser Phase müssen die groben Anforderungen konkretisiert werden. Der Vorteil beim Experience Forum ist der Zusammenhang zwischen den groben Anforderungen und der Position im Geschäftsprozess oder der Softwareanwendung, welche es betrifft. Das hilft den Interessensvertreter zu finden. Die Mehrdeutigkeit und Widersprüchlichkeit muss auch klar gestellt werden.

- Negotitaion (Verhandeln):

Hier werden die Anforderung identifiziert, sowie auch Inkonsistenzen und Konflikte aufgelöst. Zusätzlich werden die Anforderung priorisiert. Offensichtlich wird ein Fingerspitzengefühl verlangt, besonders wenn Fehlern und besonderen Merkmalen eine niedrige Priorität zugeordnet wird.

- Dokumentation:

Die Anforderungen werden in einer für den Entwickler geeigneten Form fixiert.

- Verifikation/Validierung:

Das Prüfen auf Form und Inhalt wird durch das Experience Forum unterstützt.

Anforderungsmanagement:

- Change Management (Änderungsmanagement):

  Gelöste Angelegenheiten müssen benannt oder gelöscht werden. Geänderte Probleme, die die Software oder die Geschäftsprozesse betreffen, müssen überprüft werden.

- Tracing (Verfolgbarkeit):

  Experience Forum beinhaltet nützliche Information über den Anforderungskontext. Die Verfolgbarkeit kann mit Einbezug des Experience Forums gelöst werden.

Da die Rückmeldungen erst ab der ersten funktionierenden Softwareversion aufgefangen werden können, können die aus Experience Forum folgenden Verbesserungen beim zweiten Anlauf berücksichtigt werden. Bevor aber die endgültige Version für den Anwender verfügbar wird, soll das Experience Forum an die neue Version angepasst werden. Für die Anforderungsanalyse ist es wichtig die Erfahrungen, aus denen die neuen Anforderungen abgeleitet wurden, als abgeschlossen zu markieren. Die Entwickler können normalerweise mit den gewonnenen Informationen ihre Arbeit vervollständigen.

3.3 Funktionen und Layout von Experience Forum

3.3.1 Feedbacktypen

Die Feedbackinformationen werden von den Entwicklern und Prozessmanagern verarbeitet. Dabei ist es notwendig die Einträge zu filtrieren und bestimmten Gruppen zuzuordnen. Aus diesem Grund sollte sich der Benutzer schon beim Eintragen zwischen den vier verschiedenen Arten von Feedback entscheiden. Das gewährleistet die von dem User gewünschte Zuordnung und erspart dem Forumadministrator das aufwändige Filtern.

Es handelt sich dabei um die folgenden Typen [2]:

Programmfehlerbericht (Bug Report) :

Die Benutzer berichten über Fehler, die sie im Programm finden. Das könnte in den Entwicklungsprozess einfließen.

Verbesserungsvorschlag (Feature Request) :

Benutzer beobachten ein Merkmal, das ihnen den täglichen Umgang bei der Arbeit erleichtern würde. Diese Vorschläge unterstützen dann letztlich auch den Entwicklungsprozess.

Prozessmängel und –verbesserung (Process Improvement) :

Benutzer können einen Eintrag bezüglich des Geschäftsprozesses machen. Manche Rückmeldungen betreffen den gesamten Prozess. Die Geschäftsprozessdesigner sollten solche Kommentare berücksichtigen.

Prozesserfahrung (Process Experience) :

Die Benutzer sind diejenigen, die an den Geschäftsprozessen teilnehmen. Sie begegnen bestimmten Regeln und unerwarteten Situationen. Wenn sie eine neue Situation bewältigt haben, können sie davon im Forum berichten und lassen so andere davon profitieren.

Die Entwickler verwenden die Einträge der ersten beiden Typen. Diese können dann in die weiteren Entwicklungsprozesse einfließen.

Die beiden letzten Arten adressieren sich an den Prozessdesigner, der dann beim nächsten Modellieren der Geschäftsprozesse auf die Anregungen eingehen kann. Die Prozesserfahrungen wenden sich zusätzlich auch an die anderen Anwender.

Ein weiterer Feedbacktyp ist **Handbuch (Handbook)**. Der Inhalt wird von den Entwicklern und Prozessdesignern gesetzt und wird im Forum angezeigt. Es beschreibt die Geschäftsaufgabe und erleichtert die Bearbeitung. Außerdem führen diese Einträge den Anwender durch den Geschäftsprozess.

Es stellt sich die Frage, wie man diese ganzen Informationen geschickt, Platz sparend und gleichzeitig übersichtlich für den Benutzer darstellt. Während der Entwicklung stellte sich immer wieder die Aufgabe, wie man geschickt das Forum dem Benutzer präsentiert. Denn der Anwender sollte sich nicht von einer Vielzahl von Komponenten oder Dialogen demotivieren lassen. Das Aussehen des Forums sollte ihn sogar dazu animieren etwas beizutragen. Im folgenden Text wird das Problem des Platzmangels und der Übersichtlichkeit anhand des Einfügens des Feedbacktypen erläutert. Es wurden am Anfang vier Prototypen entwickelt und dabei mit der Darstellung des Forums experimentiert.

Prototyp 1 (Abb. 5):

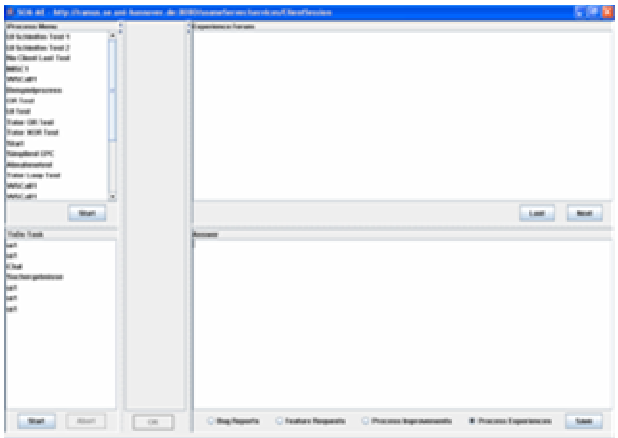

Abbildung 5: Auswahlknöpfe horizontal angeordnet

Durch die horizontale Anordnung der Auswahlknöpfe stellt sich das Forum in den Vordergrund und drängt die Aufgabenbearbeitungsfläche in den Hintergrund, wobei das die eigentliche Aufgabe ist.

Prototyp 2 (Abb. 6):

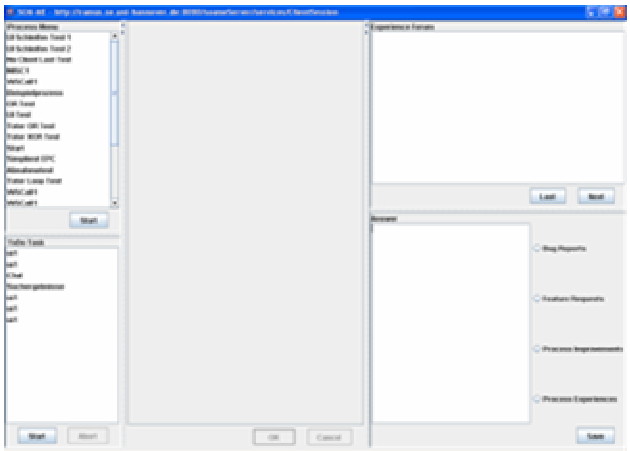

Abbildung 6: Auswahlknöpfe vertikal angeordnet

Es ist immer noch nicht die optimale Lösung. Wenn das Forum die Übersichtlichkeit behalten sollte, müsste es diese Proportionen behalten, wodurch die Arbeitsfläche verkleinert wird.

Prototyp 3 (Abb. 7):

Abbildung 7: Dialogfenster zur Auswahl des Typs

Beim Erscheinen des Dialogfensters muss der Anwender eine zusätzliche Interaktion ausführen. Das könnte ihn beim nächsten Mal davon abhalten, etwas ins Forum zu schreiben.

Prototyp 4 (Abb. 8):

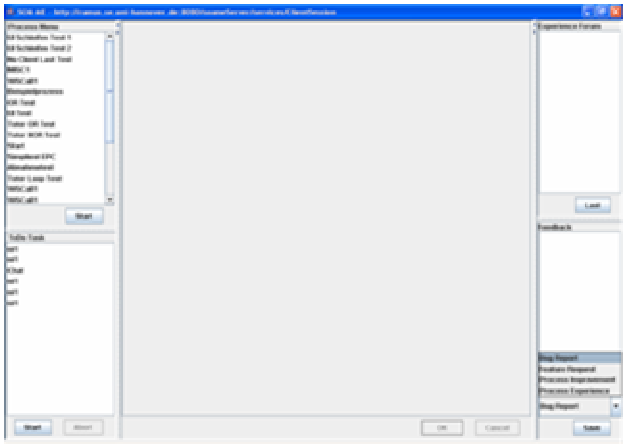

Abbildung 8: Einsatz der Auswahlliste

Durch das Einsetzen der Auswahlliste spart man Platz und der Benutzer behält die Übersicht. Er kann durch wenige Mausklicks den passenden Typ auswählen. Diese Version erscheint benutzerfreundlich und gleichzeitig Platz sparend.

3.3.2 Bewerten von Einträgen

Dem Leser wird die Möglichkeit gegeben, die Beiträge anderer User zu bewerten. Dadurch wird deutlich, welche von den Anregungen bei den Lesern die meiste Resonanz finden. Um es einfach zu gestalten und um die Reaktion der Benutzer beim Lesen am besten zu erfahren, wurde die Art der Zustimmung und Ablehnung gewählt (Abb. 9). Dabei bekommt der nächste Leser beim Sortieren der Einträge nach den Bewertungen ein besseres Bild von dem Kommentar und wird vielleicht dazu angeregt selber einen Beitrag, vielleicht auch eine Antwort, zu leisten.

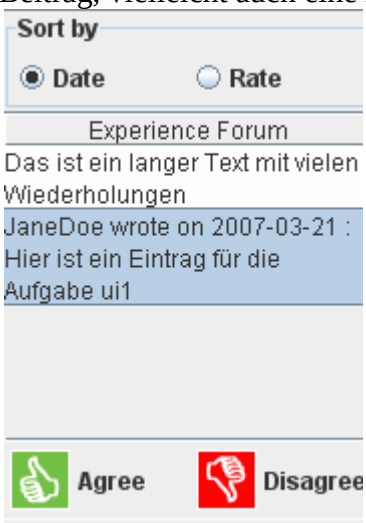

Abbildung 9: Experience Forum

3.4 Aussehen des Experience Forums

Das Forum wird durch den Platzmangel enorm eingeschränkt, die meisten Beiträge erstrecken sich jedoch über mehrere Zeilen. Um dies zu überwinden, werden nur ein paar Zeilen des Feedbacks angezeigt. Dadurch erhält der Leser genug Informationen, um zu erfahren, ob er sich für den Eintrag interessiert und weiterlesen möchte. Es trennt ihn dann nur ein Klick von dem gesamten Beitrag und den zusätzlichen Informationen(s. Abb. 9). Durch diese Methode hat der Leser eine Vielzahl von Kommentaren auf einem Blick und kann sich schließlich für einen entscheiden.

Außerdem wird versucht das Forum klein und übersichtlich zu halten. Das bedeutet, es werden nur prozessbezogenen Beiträge angezeigt. Die Informationen, die der Entwicklung dienen, erscheinen dem Leser überflüssig und sind zudem nicht hilfreich bei der Ausführung der Aufgabe.

4 Implementierung

4.1 Beschreibung der Architektur

Bei der Entwicklung des Forums stellt der Client eine Schnittstelle zwischen dem SOA-Me- System und dem Experience Forum dar (Abb. 10). Dadurch wird dem User ermöglicht durch nur einen Client an allen ihn betreffenden Geschäftsprozessen teilzunehmen und Beiträge ins Forum zu schreiben bzw. Beiträge anderer anzusehen. Es erspart dem Anwender bei der Benutzung Zeit und erleichtert die Zuordnung des Eintrags zu einer Aufgabe. Zudem stehen die Informationen durch die Verzahnung des Forums mit dem Client direkt an der Stelle zur Verfügung, wo sie dem Benutzer am meisten helfen.

Der Server ist innerhalb einer Apache Tomcat Ausführungsumgebung lauffähig und kommuniziert mit dem Client via Webservices. Webservices sind Software-Anwendungen, die mit einem Uniform Resource Identifier (URI) eindeutig identifizierbar sind und deren Schnittstellen als XML-Artefakte definiert, beschrieben und gefunden werden können [4], [5]. Dieser wiederum kommuniziert mit der Feedback-Datenbank, die sich in einem relationalem Datenbanksystem MySQL befindet, und stellt SQL-Anfragen. Dies wird von JDBC, der Datenbankschnittstelle der Java-Platform, unterstützt. Als Antwort sendet der Webservice an den Client die benötigten Informationen aus der Datenbank.

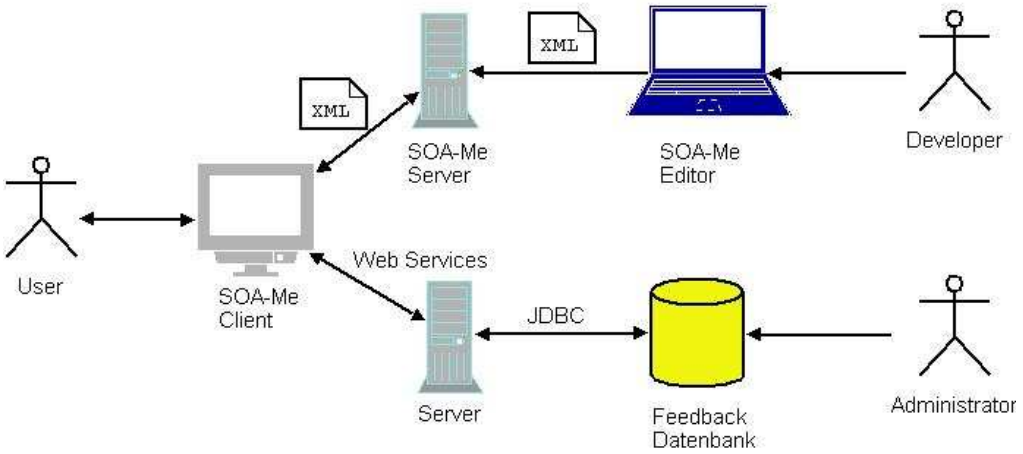

Abbildung 10: Gesamtarchitektur

4.2 Anforderungen

Anforderungen:

Bei der Entwicklung vom Experience Forum sollten folgende Anforderungen berücksichtigt werden. Sie stellen die Funktionalität und die Darstellung vom Forum sicher.

Hohe Priorität haben folgende Punkte, die das Experience Forum charakterisieren:

1. Die Erfahrungen werden über einen Webservice in einer Datenbank verwaltet.
2. Berichte werden in einer geeigneten Darstellung dem Anwender angezeigt.
3. Der Benutzer wählt den Typ seiner Rückmeldung.

Optionale Anforderungen an das Forum sind:

4. Es ist möglich einzelne Beiträge zu bewerten
5. Die Beiträge werden entweder nach den Bewertungen oder nach dem Datum sortiert. Dabei werden die bestbewerteten bzw. die aktuellsten Beiträge zuerst angezeigt

4.2.1 Funktionale Anforderungen

Use Case Diagramm:

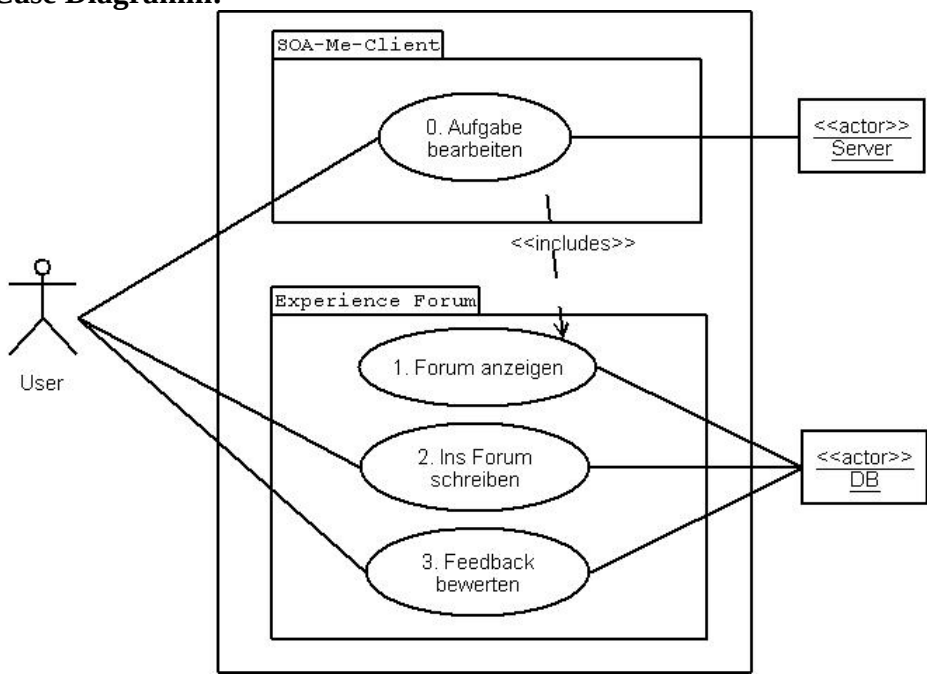

Abbildung 11: Use Case Diagramm

Im Anhang A wird das Use Case-Diagramm (Abb. 11) näher beschrieben. Das Use Case „Aufgabe bearbeiten“ findet jedoch keine nähere Betrachtung, da dieses ein bereits vorhandener Bestandteil des SOA-Me-Clients und in dessen Anforderungen bereits spezifiziert worden ist.

4.2.2 Qualitätsanforderungen

Das Experience Forum muss folgende Qualitätsaspekte beachten:

- Nützlichkeit (Usefulness):

Funktionalität der Services und wählbare Optionen

- Bedienbarkeit (Usability):

Es müssen folgende Teilaspekte von Usability wie Übersichtlichkeit und Selbstbeschreibungsfähigkeit berücksichtigt werden.

Der SOA-Me-Client spielt eine zentrale Rolle, da er eine graphische Benutzerschnittstelle bereitstellt, die jeder User bedienen können muss. Wichtig ist ein übersichtliches Aussehen und außerdem eine schnelle und einfache Bedienbarkeit.

Unter anderem sind Qualitätsaspekte wie Portabilität, Korrektheit, Stabilität und Zuverlässigkeit wichtig, da unter der Beachtung dieser Merkmale die Funktionalität gewährleistet werden kann.

4.3 Feedback-Datenbank

Zur Verwaltung von Erfahrungen wurde wie unter Abb. 12 dargestellte Datenbank entworfen:

- USER (<u>user</u>, password)

USER beinhaltet Informationen über die User. Jeder Benutzer hat einen eindeutigen Namen (user) und ein Passwort (password).

- FEEDBACK(<u>id</u>, server, p_id, t_id, author -> USER, text, type, date, done)

Ein **FEEDBACK** wird durch die Id identifiziert. Zusätzlich zum Text (text) ist der Verfasser (author) und Typ (type) bekannt. Hilfreich für die Zuordnung zu einer Aufgabe sind die Angaben zu Server, Prozess-Id und ToDo-Id gespeichert. Als zusätzliche Hinweise werden auch Datum und die Done-Markierung gespeichert.

- RATING (user -> USER, feedback_id -> FEEDBACK, rating)

RATING enthält die Bewertung der einzelnen Einträge zusammen mit den notwendigen Angaben.

Mit Hilfe der Datenbank können die Userinteraktionen verwaltet werden. Zum Beispiel werden beim Speichern eines Eintrags die Daten mit den zugehörigen Hinweisen in der Tabelle abgelegt.

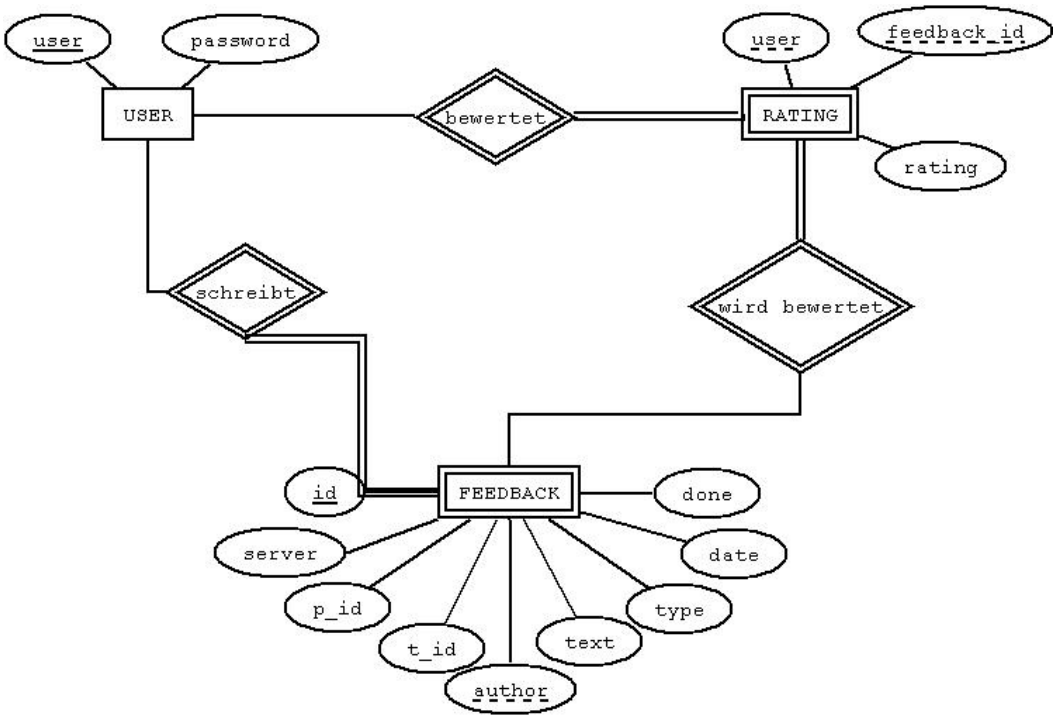

Abbildung 12: ER-Diagramm der Datenbank

4.4 Beschreibung der Webservices

Operationen, Parameter- und Rückgabetypen der Webservices sind durch die WSDL-Datei im Anhang A vollständig beschrieben. Im Folgenden wird ein Überblick über die Operationen gegeben.

Operation „setForumContent“

Bezeichnung	**Typ**	**Inhalt**
Parameter		
server	String	Serveridentifizierung
pid	String	Prozessidentifizierung
tid	String	Aufgabeidentifizierung

sort	String	Parameter, nach dem sortiert werden soll
Rückgabe		
setForumContentReturn	Feedback[1][]	Foruminhalt mit vier Einträgen: Feedback ID Text Autor Datum

Diese Operation holt sich das Feedback mit den nötigen Informationen aus der Datenbank. Die erhaltenen Daten bilden dann den Inhalt des Forums.

Operation „insertFeedback“

Bezeichnung	**Typ**	**Inhalt**
Parameter		
server	String	Serveridentifizierung
pid	String	Prozessidentifizierung
tid	String	Aufgabenidentifizierung
text	String	Feedback
author	String	Autor
type	String	Feedbacktyp (Bug Report, Feature Request, Process Improvement, Process Experience)
Keine Rückgabe		

Möchte ein Benutzer seine Erfahrungen oder Verbesserungsvorschläge ins Experience Forum eintragen, so wird diese Operation aufgerufen. Sie übergibt den Server, die Prozess-, sowie die Aufgabenidentifizierung, den eigentlichen Beitrag, den Autor und den Beitragstyp an die Datenbank. Dabei wird diesem Beitrag automatisch eine neue ID zugewiesen.

Operation „insertRateFeedback“

Bezeichnung	**Typ**	**Inhalt**
Parameter		
user	String	Angemeldeter Anwender
feedback_id	Integer	Feedback ID
rate	Integer	Bewertung
Keine Rückgabe		

Die Bewertung eines Eintrags findet mit Hilfe dieser Operation statt. Dabei werden der Datenbank die Feedback ID, der bewertende Benutzer und die

[1] Feedback ist ein komplexer Datentyp: ID : String, Text : String, Author : String, Date : Date

eigentliche Bewertung mitgeteilt. Durch das Speichern des bewertenden Benutzers wird verhindert, dass weder der Autor seinen eigenen Beitrag bewerten kann, noch andere Benutzer den gleichen Beitrag mehrmals bewerten.

4.5 Verwaltung im Experience Forum

DaimlerChrysler hat mehrere Versuche zur Entwicklung und Integration von Experience Bases in den Entwicklungsprozess unternommen [3]. Es wurde dabei schrittweise eine Experience Base entwickelt, die den Entwicklern erlauben sollte leicht ihre Erfahrungen mitzuteilen. Ferner wurde auch nach der passenden Darstellung gesucht. Dabei wurden folgende Aspekte erarbeitet, die hier auf das Forum übertragen werden:

- Am Anfang sollten potentielle Benutzer auf keinen Fall auf ein leeres Forum treffen. Es müsste am Anfang etwas Material im Forum stehen, möglicherweise ein Handbucheintrag, wie man die Aufgabe handelt.
- Dem Anwender sollte es leicht sein, seinen Beitrag ins Forum zu schreiben. Durch einfaches Design und Form sollte es gelingen nutzvolle Benutzer zu stimulieren.
- Damit manche Benutzer den Kontakt mit dem Forum nicht vermeiden, sollte der Administrator auf doppelte Einträge achten und diese ggf. löschen. Diese machen sonst das Lesen fade und mühsam.
- Es muss erkennbar sein, dass die Einträge auch verarbeitet und die Prozesse zu den Wünschen von Nutzern verbessert werden. Um die Priorität bestimmen zu können, wird dem User die Gelegenheit gegeben einem Eintrag zuzustimmen oder ihn abzulehnen.

Es ist ausschlaggebend, dass die niedergeschriebenen Erfahrungen auch verwendet werden. Das bedeutet für das Experience Forum, dass dieses Vorgehen dem Anwender deutlich veranschaulicht werden muss. So bildet sich ein ständiger Austausch zwischen dem Entwicklerteam und den Designern auf der einen Seite und dem Anwender auf der anderen Seite. Dabei benutzen alle Interessensvertreter die gleiche Datenbank wie in der Abbildung 13 gezeigt wird.

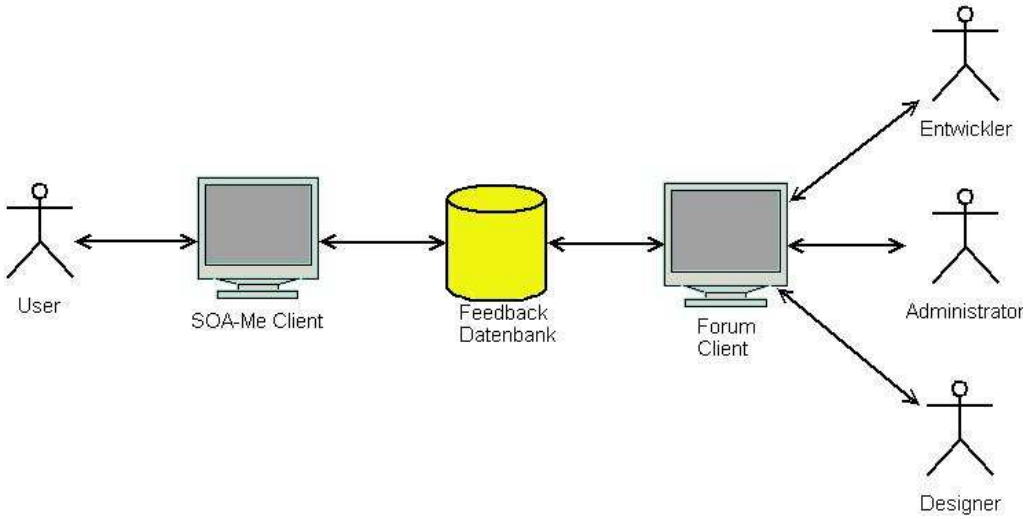

Abbildung 13: Zugriffe auf die Feedback Datenbank

Es ist jedoch klar, dass für das Entwicklerteam, die Designer und den Administrator der SOA-Me-Client zur Einsicht und Verwaltung der Inhalte der Datenbank nicht geeignet ist, weil sie nicht an den Geschäftprozessen teilnehmen. Zudem ist für sie störend, dass die Erfahrungen nur selektiert zu den Geschäftsaufgaben einzusehen sind. Stattdessen bräuchten sie eine Überblicksdarstellung und die Möglichkeit, einfach und schnell Erfahrungen bearbeiten zu können. Aus diesem Grund wurde ein einfacher Client wie in Abbildung 14 entworfen, der eine geeignete Sichtweise auf den Inhalt der Datenbank bietet.

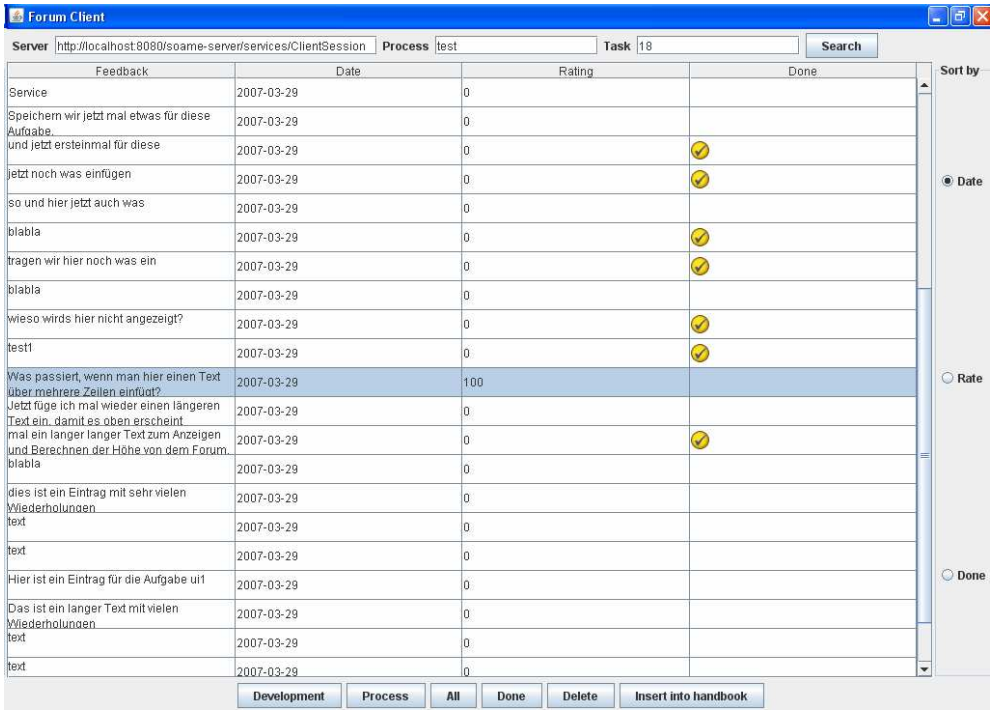

Abbildung 14: Client zur Verwaltung im Experience Forum

Die funktionalen Anforderungen aus der Abbildung 15 sind notwendig um die Datenbank sinnvoll zu verwalten und um schnell und einfach die notwendigen Inhalte anzuzeigen.

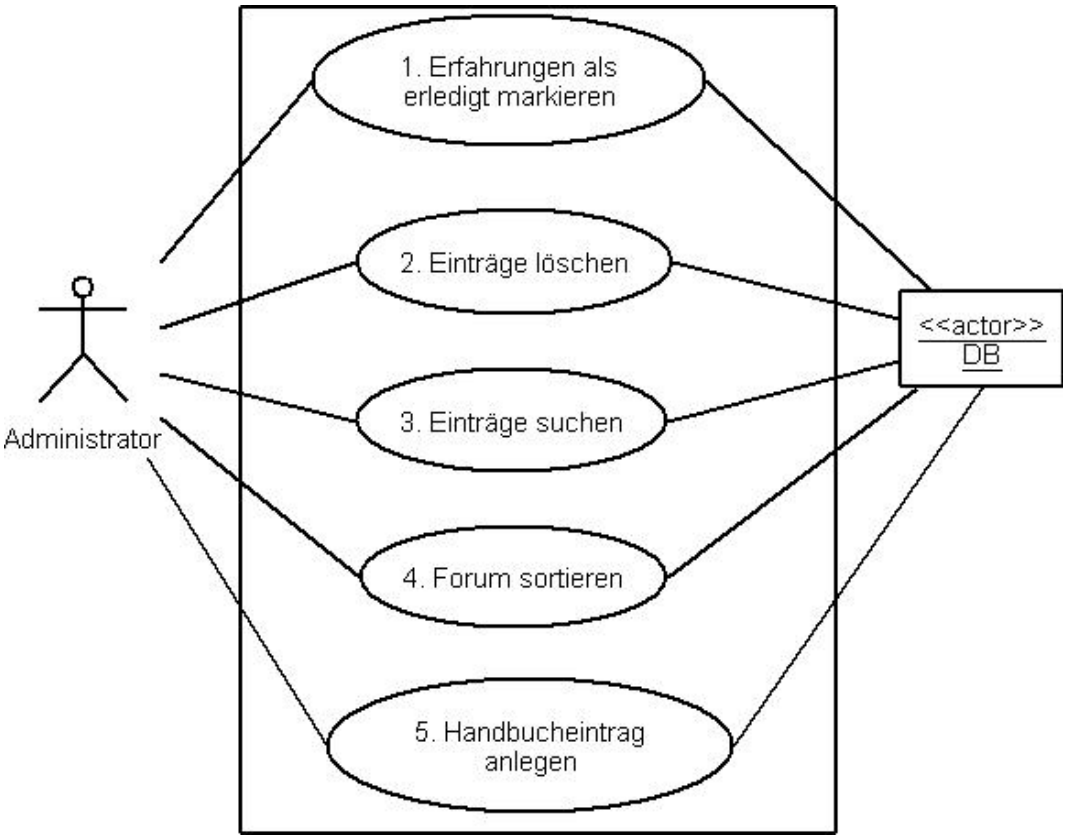

Abbildung 15: Use Case Diagramm für den Forum Client

Die Anwendergruppen von diesem Client sind Entwickler, Prozessdesigner, On-Site Customer und die Administratoren. Sie können den Inhalt der Datenbank einsehen und schließlich für ihre Zwecke bearbeiten. Zusätzlich können sie einen ein Handbuch für die Anwender anlegen. Um einzelne Einträge selektieren zu können, bietet sich ihnen die Möglichkeit einen expliziten Server, Prozess und Aufgabe zur Suche anzugeben. Zur zusätzlichen Orientierung kann der Inhalt auch nach Datum, Bewertung oder Bearbeitungsstatus sortiert werden. Die Erfahrungen werden als bearbeitet markiert, sobald sie für die Interpretation der Anforderungen erfasst wurden. Sie dienen später der Archivierung von Feedback und werden nicht mehr im Forum erscheinen. Weitere Aufgabe ist es doppelte oder überflüssige Einträge zu löschen.

5. Anwendungsszenario

Dieser Abschnitt soll noch mal die Arbeit mit dem Experience Forum veranschaulichen.

Ein Anwender arbeitet gerade an der Prozessaufgabe „Suchdienst auswählen und Suchwort eingeben“. Durch den Handbucheintrag im Forum wird noch mal klar, was der Anwender bei dieser Aufgabe tun soll. Aber er möchte seine Suche noch einschränken. Also schreibt er einen Eintrag mit der Verbesserung ins Forum (Abb. 16).

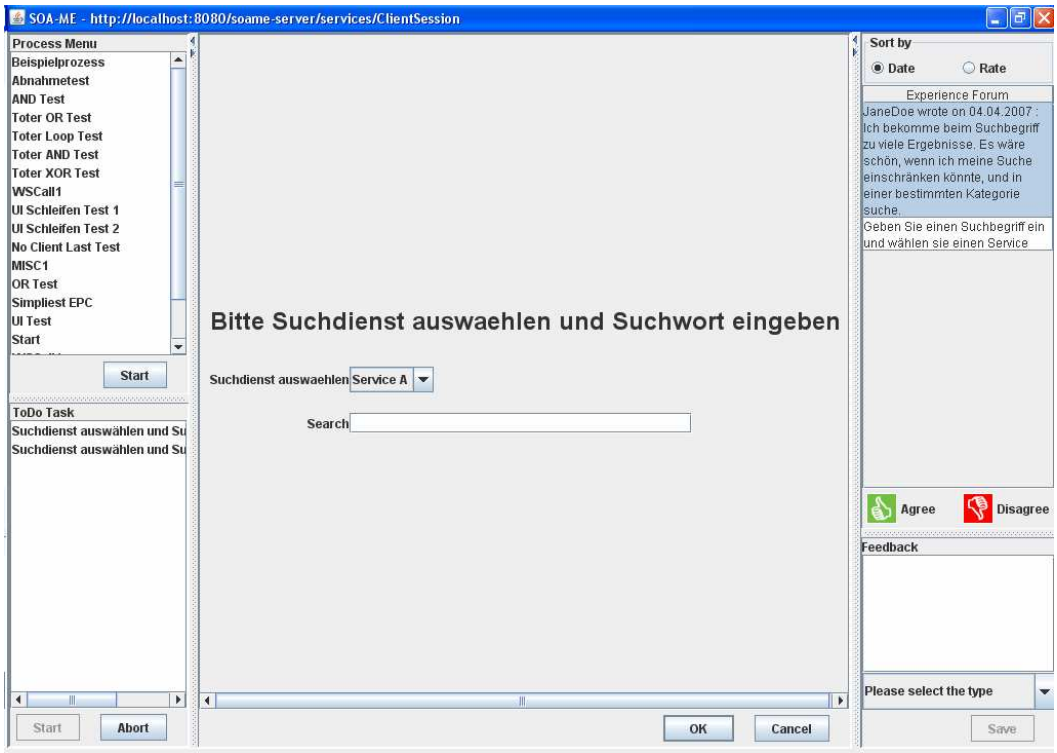

Abbildung 16: Eintrag zum Geschäftsprozess

Ein anderer Benutzer hat das gleiche oder ein ähnliches Problem bei dieser Geschäftsaufgabe. Er war gerade auf der Suche nach der passenden Lösung im Forum und las die Einträge anderer, als ihm dieser Beitrag ins Auge fiel. Sofort stimmt er diesem zu. So könnte es mehreren Anwendern ergangen sein, also schreiben sie auch ins Forum oder sie bewerten die Beiträge anderer.

Vor der Entstehung einer neuen Version des Geschäftsprozesses liest sich der Designer die gesammelten Feedbacks durch (Abb. 17). Aus dem obigen Eintrag und den dazugehörigen Bewertungen entnimmt er, dass viele User sich noch ein zusätzliches Feld wünschen, um ihre Suche einzugrenzen. Er modelliert diese

Aufgabe um und fügt ein neues Feld hinzu, die den Anwendern die Möglichkeit gibt noch eine Kategorie für ihre Suche anzugeben.

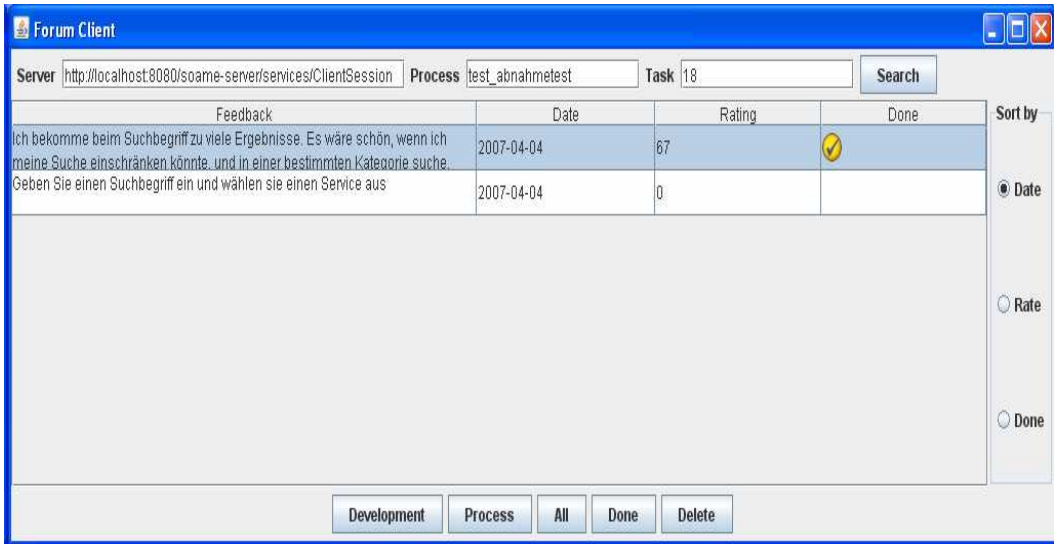

Abbildung 17: Bearbeitung des Feedbacks

Da dieser Eintrag schon bearbeitet wurde, wird er markiert und erscheint nicht mehr im Forum.

Bei der neuen Version erscheint dann noch ein zusätzliches Eingabefeld (Abb. 18). Die Anwender bemerken die Änderung und überzeugen sich vom Zweck des Forums.

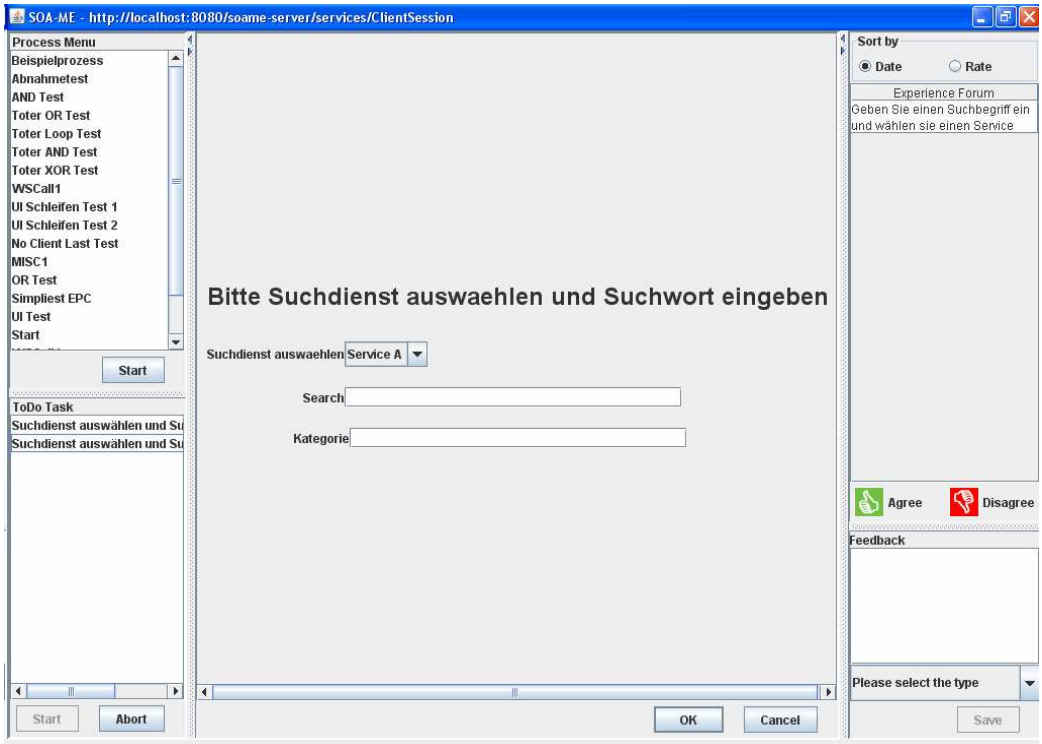

Abbildung 18: Geänderte Version der Geschäftsaufgabe

6. Zusammenfassung und Ausblick

Softwareanwendungen werden gerade in großen Unternehmen oft von einem Entwicklerteam ohne den direkten Bezug zu den Endverbrauchern geschrieben. Dabei könnten die Benutzer einen wichtigen Beitrag zur Benutzbarkeit der Software leisten.

Ziel dieser Arbeit war es eine Lösung zu entwickeln, die eine direkte Verbindung zwischen den Entwicklern, Geschäftsdesignern und der Benutzergemeinschaft herstellt. Dazu bietet das Experience Forum eine Lösung: Aus den gewonnenen Erfahrungen im Experience Forum entstehen Verbesserungsansätze, die in der Entwicklung umgesetzt werden können. Das Forum erlaubt den Austausch von Informationen und Erfahrungen direkt über den SOA-Me-Client. Außerdem kann es direkt während der Arbeit von den Benutzern verwendet werden und bietet die Möglichkeit zum Abrufen und Mitteilen von Erfahrungen. Mit Hilfe der Einbindung der Information in den Geschäftsprozess wurden das Abrufen der Inhalte und das manuelle Zuordnen erleichtert.

Neben der Integration des Forums in den SOA-Me-Client entstand auch ein Client zur Verwaltung der Einträge. Es erlaubt den in den Entwicklungsprozess bezogenen Personen einen einfachen und schnellen Zugriff auf das Feedback der Verbraucher. Der Inhalt kann anschließend leicht von ihnen genutzt und zu neuen Anforderungen umgesetzt werden.

Als nächster Schritt sollte das Experience Forum in einer breiten Benutzergemeinschaft eingesetzt und getestet werden. Dabei sollte nicht nur der entworfene Client näher betrachtet werden, sondern auch der Kreislauf, der sich zwischen den Entwicklern, Designern und den Anwendern bildet, erprobt werden.

Literaturverzeichnis

[1] Daniel Lübke and Kurt Schneider, *Leveraging Feedback on Processes in SOA Projects*. In Software Process Improvement, 13th European Conference, EuroSPI 2006, p. 194-2005

[2] Daniel Lübke and Eric Knauss, *Dealing with User Requirements and Feedback in SOA Projects*, In Proceedings of the Workshop on Software Engineering Methods for Service-Oriented Architecture (SEMSOA), 2007.

[3] Schneider, K. and T. Schwinn, *Maturing Experience Base Concepts at DaimlerChrysler*, Software Process Improvment and Practice, 2001.6: p. 85-96

[4] W3C, *Web Services Architecture Requirements*, http://www.w3.org/TR/wsa-reqs/, zugegriffen am 04.04.2007

[5] *Wikipedia*, http://de.wikipedia.org/wiki, zugegriffen am 05.04.2007

Abbildungsverzeichnis

Abbildung 1: SOA-Me-System Architektur 4
Abbildung 2: SOA-Me Client 5
Abbildung 3:Informationsfluss in der Softwareentwicklung 6
Abbildung 4: Phasen der Anforderungsanalyse 9
Abbildung 5: Auswahlknöpfe horizontal angeordnet 11
Abbildung 6: Auswahlknöpfe vertikal angeordnet 12
Abbildung 7: Dialogfenster zur Auswahl des Typs 12
Abbildung 8: Einsatz der Auswahlliste 13
Abbildung 9: Experience Forum 13
Abbildung 10: Gesamtarchitektur 15
Abbildung 11: Use Case Diagramm 16
Abbildung 12: ER-Diagramm der Datenbank 18
Abbildung 13: Zugriffe auf die Feedback Datenbank 21
Abbildung 14: Client zur Verwaltung im Experience Forum 21
Abbildung 15: Use Case Diagramm für den Forum Client 22
Abbildung 16: Eintrag zum Geschäftsprozess 23
Abbildung 17: Bearbeitung des Feedbacks 24
Abbildung 18: Geänderte Version der Geschäftsaufgabe 24

Anhang A: Use Case Beschreibungen

Use Case Nr. 1	**Forum anzeigen**
Umfeld	Am PC mit Bildschirm und Tastatur und Internetanschluss
Systemgrenzen	Alle für das Forum benötigten Informationen bekommt der Client von der Datenbank
Ebene	Hauptaufgabe
Hauptakteur	Client
Stakeholder und Interessen	User Liest die Meinung anderer User
Voraussetzung	Programm gestartet; es besteht eine Verbindung zur Datenbank
Garantien	-
Erfolgsfall	Die zu der Todo-Aufgabe zugehörende Information wird angezeigt
Auslöser	User arbeitet eine Todo-Aufgabe ab
Beschreibung	1. User arbeitet eine Todo-Aufgabe ab. 2. Client ruft von der Datenbank die dazugehörige Information ab. 3. Die Information wird ohne Userinteraktion angezeigt
Erweiterungen	2a WENN Datenbank nicht verfügbar DANN bleibt das Fenster leer
Technologien	-

Use Case Nr 2	**Ins Forum schreiben**
Umfeld	Am PC mit Bildschirm und Tastatur und Internetanschluss
Systemgrenzen	Die vom User eingegebenen Daten werden an die Datenbank geschickt
Ebene	Hauptaufgabe
Hauptakteur	User
Stakeholder und Interessen	User Teilt seine Meinung mit Entwickler Erkennt die Schwächen des Programms
Voraussetzung	Programm gestartet; es besteht eine Verbindung zur Datenbank
Garantien	-
Erfolgsfall	Die Rückmeldung wurde in die Datenbank eingetragen
Auslöser	User füllt das vorgesehene Fenster aus und bestätigt

Beschreibung	1. User füllt das vorgesehene Fenster aus und bestätigt User bestätigt die Eingabe. 2. Client sendet die Eingabe an die Datenbank
Erweiterungen	2a WENN die Eingabe nicht erfolgreich war DANN erscheint eine Fehlermeldung
Technologien	-

Use Case Nr 3	**Feedback bewerten**
Umfeld	Am PC mit Bildschirm und Tastatur und Internetanschluss
Systemgrenzen	Die vom User eingegebenen Daten werden an die Datenbank geschickt
Ebene	Nebenaufgabe
Hauptakteur	User
Stakeholder und Interessen	User Priorisiert die Einträge Entwickler Erkennt die Priorität der Einträge
Voraussetzung	Programm gestartet; es besteht eine Verbindung zur Datenbank
Garantien	-
Erfolgsfall	Die Bewertung wurde in die Datenbank eingetragen
Auslöser	User drückt den Button Agree oder Disagree
Beschreibung	1. User betätigt einen der Bewertungsbuttons 2. Client sendet die Bewertung an die Datenbank
Erweiterungen	-
Technologien	-

Anhang B: Webservice Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://db"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://db" xmlns:intf="http://db"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<!--WSDL created by Apache Axis version: 1.3
Built on Oct 05, 2005 (05:23:37 EDT)-->
 <wsdl:types>
  <schema elementFormDefault="qualified"
targetNamespace="http://db"
xmlns="http://www.w3.org/2001/XMLSchema">
   <element name="setForumContent">
    <complexType>
     <sequence>
      <element name="server" type="xsd:string"/>
      <element name="pid" type="xsd:string"/>
      <element name="tid" type="xsd:string"/>
      <element name="sort" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="setForumContentResponse">
    <complexType>
     <sequence>
      <element maxOccurs="unbounded" name="setForumContentReturn"
type="impl:Experience"/>
     </sequence>
    </complexType>
   </element>
   <complexType name="Experience">
    <sequence>
     <element name="author" nillable="true" type="xsd:string"/>
     <element name="date" nillable="true" type="xsd:dateTime"/>
     <element name="id" type="xsd:int"/>
     <element name="text" nillable="true" type="xsd:string"/>
    </sequence>
   </complexType>
   <element name="insertFeedback">
    <complexType>
     <sequence>
      <element name="server" type="xsd:string"/>
      <element name="pid" type="xsd:string"/>
      <element name="tid" type="xsd:string"/>
      <element name="text" type="xsd:string"/>
      <element name="author" type="xsd:string"/>
      <element name="type" type="xsd:string"/>
     </sequence>
    </complexType>
   </element>
   <element name="insertFeedbackResponse">
    <complexType/>
```

```
   </element>
   <element name="insertRateFeedback">
    <complexType>
     <sequence>
      <element name="user" type="xsd:string"/>
      <element name="feedback_id" type="xsd:int"/>
      <element name="rate" type="xsd:int"/>
     </sequence>
    </complexType>
   </element>
   <element name="insertRateFeedbackResponse">
    <complexType/>
   </element>
   <complexType name="DoubleException">
    <sequence/>
   </complexType>
   <element name="fault" type="impl:DoubleException"/>
  </schema>
 </wsdl:types>

   <wsdl:message name="insertRateFeedbackResponse">

      <wsdl:part element="impl:insertRateFeedbackResponse"
name="parameters"/>

   </wsdl:message>

   <wsdl:message name="insertRateFeedbackRequest">

      <wsdl:part element="impl:insertRateFeedback"
name="parameters"/>

   </wsdl:message>

   <wsdl:message name="DoubleException">

      <wsdl:part element="impl:fault" name="fault"/>

   </wsdl:message>

   <wsdl:message name="insertFeedbackResponse">

      <wsdl:part element="impl:insertFeedbackResponse"
name="parameters"/>

   </wsdl:message>

   <wsdl:message name="insertFeedbackRequest">

      <wsdl:part element="impl:insertFeedback" name="parameters"/>

   </wsdl:message>

   <wsdl:message name="setForumContentResponse">
```

```
      <wsdl:part element="impl:setForumContentResponse"
name="parameters"/>

   </wsdl:message>

   <wsdl:message name="setForumContentRequest">

      <wsdl:part element="impl:setForumContent" name="parameters"/>

   </wsdl:message>

   <wsdl:portType name="DbConnection">

      <wsdl:operation name="setForumContent">

         <wsdl:input message="impl:setForumContentRequest"
name="setForumContentRequest"/>

         <wsdl:output message="impl:setForumContentResponse"
name="setForumContentResponse"/>

      </wsdl:operation>

      <wsdl:operation name="insertFeedback">

         <wsdl:input message="impl:insertFeedbackRequest"
name="insertFeedbackRequest"/>

         <wsdl:output message="impl:insertFeedbackResponse"
name="insertFeedbackResponse"/>

      </wsdl:operation>

      <wsdl:operation name="insertRateFeedback">

         <wsdl:input message="impl:insertRateFeedbackRequest"
name="insertRateFeedbackRequest"/>

         <wsdl:output message="impl:insertRateFeedbackResponse"
name="insertRateFeedbackResponse"/>

         <wsdl:fault message="impl:DoubleException"
name="DoubleException"/>

      </wsdl:operation>

   </wsdl:portType>

   <wsdl:binding name="DbConnectionSoapBinding"
type="impl:DbConnection">

      <wsdlsoap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>

      <wsdl:operation name="setForumContent">
```

```
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="setForumContentRequest">
            <wsdlsoap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="setForumContentResponse">
            <wsdlsoap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="insertFeedback">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="insertFeedbackRequest">
            <wsdlsoap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="insertFeedbackResponse">
            <wsdlsoap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="insertRateFeedback">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="insertRateFeedbackRequest">
            <wsdlsoap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="insertRateFeedbackResponse">
            <wsdlsoap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="DoubleException">
            <wsdlsoap:fault name="DoubleException" use="literal"/>
```

```
        </wsdl:fault>

      </wsdl:operation>

   </wsdl:binding>

   <wsdl:service name="DbConnectionService">

      <wsdl:port binding="impl:DbConnectionSoapBinding"
name="DbConnection">

        <wsdlsoap:address
location="http://localhost:8080/DBService1/services/DbConnection"/>

      </wsdl:port>

   </wsdl:service>

</wsdl:definitions>
```

Anhang C: Installation und Konfiguration

Auf der beiliegenden CD befinden sich die zur Installation benötigten Dateien.
Installation:

1. Zum Ausführen der Anwendung wird Tomcat benötigt. Der Installationspfad von Tomcat wird im Folgenden als <catalina_home> bezeichnet.
2. Kopieren sie soame-server.war ins <catalina_home>/webapps Verzeichnis.

3. Kopieren sie DBService1.war ins <catalina_home>/webapps Verzeichnis.

4. Als nächstes muss das context.xml – File angepasst werden. Zu ändernde Einstellungen:
   - Username Der Benutzername von der Datenbank
   - Password Das Passwort zu der Datenbank
   - url Die Url, unter der die Datenbank verfügbar ist

5. Erstellen sie eine Datenbank mit Hilfe von feedback.sql-Datei.

6. Starten sie Tomcat und client.jar. Geben Sie als Server http://localhost:8080/soame-server/services/ClientSession an.

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit und die zugehörige Implementierung selbständig verfasst und dabei nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Hannover, 10. April 2007

Sabina Grimm