Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Philipp Prögel

Dienstkomposition für kooperatives Arbeiten in der Lehre

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Philipp Prögel

Dienstkomposition für kooperatives Arbeiten in der Lehre

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Martin Becke
Zweitgutachter: Prof. Dr. Thomas Lehmann

Eingereicht am: 15. Juni 2016

Philipp Prögel

Thema der Arbeit
Dienstkomposition für kooperatives Arbeiten in der Lehre

Stichworte
Web-Dienst, Dienstkomposition

Kurzzusammenfassung
Die heutige Lehr- und Arbeitswelt entwickelt sich immer stärker in eine Richtung der agilen Gruppen- und Projektarbeit. Beide Welten haben das gemeinsame Ziel, die Herausforderungen in der Kooperation möglichst effizient zu lösen. Einen besonderen Einfluss auf den Gesamterfolg hat hier die Organisation [1] innerhalb des Teams.

Um die Zusammenarbeit zu erleichtern, sind verschiedene Web-Dienste für die Optimierung der Organisation[1] entstanden. Insbesondere verschiedene Ansätze aus den Service-orientierten Architekturen (SOA) werden in diesem Kontext immer erfolgreicher. So können zum Beispiel Dateien von überall aus der Welt gemeinsam aktiv bearbeitet werden. Auch Anwendungen die auf Echtzeitkommunikation basieren, wie textbasierte Chats oder interaktive Videokonferenzen, sind ein wichtiger Teil des Angebots. Gemeinsam ist all diesen Diensten, das die Kommunikation und die Organisation innerhalb einer Gruppe deutlich vereinfacht wird. Sie lösen nicht das Problem, helfen aber bei der Problemlösung.

Doch diese Vielfalt an Diensten stellt auch eine Herausforderung für die einzelnen Gruppenmitglieder dar. Üblich ist, dass jeder dieser Dienste unabhängig für sich alleine angeboten wird, aber nicht unabhängig in der Nutzung zu sehen ist. So hat der Anwender eine noch wenig diskutierte aber wichtige Herausforderung in der Komposition dieser Dienste zu erfüllen. Beispielhaft muss jeder Nutzer für sich organisieren, ob und wann er pro Dienst eine Applikation startet oder eine Webseite besucht. Dies ist eine nicht zu unterschätzende Aufgabe, die nicht nur mit der Zeit für die Synchronisation der Dienste zu beschreiben ist. Automatisierte Mechanismen zur Unterstützung sind bisher noch nicht etabliert. Diese Arbeit soll einen ersten Beitrag für zukünftige Lösungen anbieten mit der Umsetzung einer unterstützenden Middleware.

Zusammengefasst wird im Rahmen dieser Bachelorarbeit eine erste Middleware für die Dienstkomposition zum kooperativem Arbeiten für Studenten entwickelt, diegeeignete Dienste für

kooperatives Arbeiten in der Lehre zusammenfasst. Hierfür werden insbesondere externe Web-Dienste auf ihre Fähigkeit für die Dienstkomposition innerhalb eines Framworks untersucht. Die Betrachtung der Schnittstellen, also die Application-Programming-Interfaces (API), spielen eine besondere Rolle und bedürfen daher einer besonderen Betrachtung. Auch weitere technische, dienstabhängige Kriterien der Dienstkomposition sind zu identifizieren und als Kriterienkatalog für die zu erstellende Middleware innerhalb dieser Arbeit bereitzustellen.

Eine weitere Designanforderung in der Entwicklung ist der Aufbau einer möglichst generischen Lösung. Primär bedeutet dies Gemeinsamkeiten der Dienste zu identifizieren und architektonisch zu abstrahieren. Hier spielt auch der Einsatz moderner Webtechnologien in einem systemadaptiven Anwendungsszenario eine besondere Rolle, um eine ressourcensparende und skalierende Lösung zu entwickeln.
Basis für die Auswahl der Dienste wird die Auswertung einer Studentenbefragung bieten, die auch im Rahmen dieser Arbeit erstellt werden wird.
Philipp Prögel

Title of the paper
Servicecomposition for cooperative work

Keywords
service, servicecomposition

Abstract
English abstract goes here TODO...

Inhaltsverzeichnis

Listings

1 Einleitung

1.1 Motivation

Aufgaben und Arbeiten während des Studiums werden immer öfter in Gruppen bearbeitet. Dabei hat die Kooperation innerhalb der Gruppe einen großen Einfluss auf das Ergebnis. Fehlende Kommunikation, planloses Vorgehen und keine Arbeitseinteilung sind Hinweise auf eine ungenügende Organisation innerhalb der Gruppe.

Um dem entgegen zu wirken wurden verschiedene Web-Dienste für das kooperative Arbeiten entwickelt. Diese werden in folgende Kategorien eingeteilt: Die mit am weitesten verbreiteten Arten von Web-Diensten sind die Cloud-Storage-Dienste [1, 5]. Dabei werden Dateien online gespeichert und über das Internet zugänglich gemacht. Dadurch können Dateien einfach mit Gruppenmitgliedern geteilt werden. Eine weitere wichtige Kategorie an Web-Diensten für kooperatives Arbeiten bilden die Instant-Messaging-Dienste [2, 80]. Mithilfe dieser können sich Gruppen in Echtzeit austauschen und aktuelle Ereignisse besprechen. Vor allem in der Informatik spielen die Web-Dienste für Versionsverwaltung eine elementare Rolle. Sie ermöglichen die Koordinierung von mehreren Entwicklern an einer Datei und die Protokollierung von Änderungen, wodurch nachverfolgt werden kann, welches Gruppenmitglied etwas geändert hat. Angelehnt an die bereits angesprochenen Cloud-Storage-Dienste existieren auch die Document Collaboration-Dienste [3, 12]. Diese erlauben eine gleichzeitige und gemeinsame Bearbeitung von Dokumenten.

Alle diese Dienste können auch mobil und dadurch flexibel genutzt werden. Der Grund dafür ist die große Verbreitung von Smartphones und damit im Zusammenhang stehend das Angebot mobiler Versionen der einzelnen Dienste. Die Infrastruktur für kooperatives Arbeiten ist somit vorhanden.

1.2 Problemstellung

Die kooperative Zusammenarbeit in der Gruppe bedarf einer Vielzahl an Diensten. Dabei steht der Anwender vor mehreren Problemen. Er muss für jeden Dienst eine Webseite besuchen oder die Applikation auf seinem Gerät starten um dessen Funktionen zu nutzen. Als Geräte kommen hierbei Computer,Laptops,Smartphones oder Tablets in Frage, die allein bei ihrer Benutzung Zeit kosten. Außerdem bestehen keine sinnvollen Automatismen für die Verknüpfung von Diensten, weshalb der Anwender die Daten zwischen den Diensten eigenständig synchronisieren muss.

Es stellt sich die Frage, wie diese Probleme gelöst werden können. Eine mögliche Antwort darauf ist die Dienstkomposition. Dabei werden verschiedene Funktionen von mehreren Diensten miteinander verbunden. Im Zuge dessen können Verknüpfungen zwischen den Diensten geschaffen werden, welche in der autonomen Ausführung der einzelnen Dienste nicht existent sind.
Es könnte beispielsweise bei jeder Änderung an einer Datei eine Nachricht per Messenger-Dienst verschickt werden, um das Team über aktuelle Entwicklungen zu informieren. Dateien aus einem Cloud-Storage-Dienst könnten an einen Document Collaboration-Dienst weitergeleitet werden, wodurch die gemeinsame Bearbeitung an der Datei ermöglicht wird. Es sind also Verknüpfungen zwischen unterschiedlichen Dienstarten vorstellbar, die allesamt die Kooperation und Organisation innerhalb der Gruppe fördern und somit bei der Problemlösung helfen.

1.3 Zielsetzung

Ziel der vorliegenden Ausarbeitung ist die Erstellung einer Dienstkomposition für Web-Dienste in der Lehre. Dabei soll der Anwendungsbereich der Dienste die Zusammenarbeit und Organisation innerhalb einer Gruppe enthalten.
Um die relevanten Web-Dienste zu identifizieren, wird im Rahmen dieser Arbeit eine Umfrage erstellt, bei der die Studenten der HAW Hamburg im Fachbereich Informatik befragt werden. Des Weiteren werden Anforderungen an die Dienstkomposition aus den Umfrageergebnissen abgeleitet. Ein Kriterienkatalog für Zusammensetzbarkeit von Web-Diensten ist ebenfalls Teil dieser Ausarbeitung. Dadurch wird festgestellt, ob sich ein Dienst für die Komposition eignet oder nicht.

Für die Dienstkomposition wird zunächst eine *Middleware* konzipiert und implementiert.

Die *Middleware* hat die Aufgabe einzelne Funktionen der Web-Dienste bereitzustellen. Die benötigten Daten werden durch verschiedene Web-Schnittstellen von den Web-Diensten zur Verfügung gestellt. Eine besondere Designvorgabe an die *Middleware* ist eine wiederverwendbare und plattformunabhängige Lösung. Im Zuge dessen können zukünftige Arbeiten über die Dienstkomposition auf die *Middleware* zurückgreifen und ihre Arbeit darauf aufbauen.

Abschließend wird mithilfe der implementierten *Middleware* eine Webseite entwickelt. Auf dieser sollen die Funktionen der Dienste, die Teil der *Middleware* sind, bereitgestellt werden.

2 Grundlagen

In diesem Kapitel werden wichtige Begrifflichkeiten im Zusammenhang mit dieser Ausarbeitung beschrieben. Dabei sollen Abgrenzungen zu bestehenden Definitionen getroffen und die Bedeutung der Begriffe beschrieben werden.

2.1 Dienst

Um im nachfolgenden Abschnitt den Begriff Web-Dienst definieren zu können muss zunächst die übergeordnete Kategorie Dienst für sich betrachtet werden. Ein Dienst beschreibt in der Informatik zumeist ein in sich geschlossenes System, welches zusammenhängende Funktionen über ein Themenfeld bündelt und mithilfe einer Schnittstelle zur Verfügung stellt.
Dienste können in unterschiedliche Arten beziehungsweise Aufgabenfelder eingeteilt werden. Zum Beispiel Systemdienste, Web-Dienste oder auch Netzwerkdienste. Dabei bestimmt die Art des Dienstes auf welche Weise die Schnittstelle erreichbar ist. Ein Systemdienst bietet seine Schnittstelle nur für das Betriebssystem an. Ein Netzwerkdienst für das gesamte Netzwerk. Und ein Web-Dienst ist von überall durch das Internet zu erreichen.
Nun können allgemeine Aussagen getroffen werden, die jeder Dienst erfüllen muss um als solcher erkannt zu werden:

- Der Dienst benötigt einen Namen oder *Identifier* um erreichbar zu sein. Falls der Dienst mit mehreren anderen Diensten in einem System gekoppelt wird, muss der Name oder *Identifier* eindeutig sein.
- Es wird eine Schnittstelle benötigt. Über diese Schnittstelle werden die Funktionen des Dienstes bereitgestellt. Innerhalb der Schnittstelle müssen die Eingabe-, Ausgabe- und Fehlerparameter als auch die Nachrichtentypen für die jeweiligen Funktionen beschrieben werden.
- Jeder Dienst besitzt einen Anwendungsbereich oder *Scope*. Dieser kann von der Applikation über das System bis hin zum Internet reichen.

2.2 Web-Dienst

Es stellt sich nun die Frage was einen Dienst zu einem Web-Dienst macht. Web-Dienste oder auch *Webservices* werden auf unterschiedliche Weisen definiert:

- Ït exposes its features programmatically over the Internet using standard Internet languages and protocols"[4, 2]
- Ït can be implemented via a self-describing interface based on open Internet standards (e.g., XML interfaces which are published in a network-based repositories)"[4, 2]

Die angesprochenen Standards für die Entwicklung der Web-Dienste sind Simple Object Access Protocol (SOAP) und Representational State Transfer (REST) [5]. Durch diese Definitionen sind Dienste die ihre Schnittstellen mit anderen Protokollen anbieten ausgeschlossen. Beispiele für andere Protokolle sind zum Beispiel Advanced Message Queuing Protocol (AMQP) oder Google Remote Procedure Call (GRPC). Aus diesem Grund wird in der vorliegenden Ausarbeitung der Begriff Web-Dienst breiter definiert. Für die Identifikation eines Web-Dienstes werden folgende notwendige Eigenschaften aufgestellt:

- Der Dienst erfüllt die oben eingeführten Eigenschaften eines Dienstes.
- Der *Scope* des Dienstes ist im Internet anzusiedeln.
- Die Schnittstelle des Dienstes ist öffentlich erreichbar und für die Bereitstellung der Dienstfunktionen vorgesehen.

2.3 Dienstkomposition

Nachdem Dienste und Web-Dienste definiert sind, muss diskutiert werden, wann eine Zusammenführung von Diensten eine Dienstkomposition darstellt. Wenn der Zusammenschluss von mehreren Diensten einen neuen komplexen Dienst erschafft, ist eine Dienstkomposition entstanden. Wichtig ist hierbei, dass die Dienste der Dienstkomposition miteinander agieren oder auf Aktionen einzelner Dienste reagieren.
Eine Dienstkomposition kann nun aus zwei Perspektiven betrachtet werden. Die erste Perspektive ist die des Anwenders. Der Anwender nutzt eine Menge an Diensten in einer Anwendung um ein Problem zu lösen. Die zweite Perspektive ist die des Entwicklers. Der Entwickler hat mithilfe einer Komposition von Diensten eine Anwendung entwickelt. Dabei muss die resultierende Anwendung aus Sicht des Anwenders keine Dienstkomposition darstellen.
Die Dienstkomposition kann entweder mit einer Orchestrierung oder mit einer Choreographie

entwickelt werden [6]. Bei der Orchestrierung wird ein zentraler Koordinator (*Orchestrator*) eingesetzt, der die einzelnen Dienste steuert und die Kommunikation reguliert. Bei der Choreographie kommunizieren die Dienste untereinander und beschreiben ihre Interaktionen eigenständig.

3 Middleware

3.1 Requirements Engineering

3.1.1 Umfrage

3.1.2 Funktionale Anforderungen

3.1.3 Nichtfunktionale Anforderungen

3.2 Technologie

3.2.1 Möglichkeiten

3.2.2 Auswahl

3.3 Analyse und Design

3.4 Architektur

4 Implementation - lessons learned

4.1 Besondere Herausforderungen

5 Anwendung der Middleware

5.1 Validierung der Anwendungsfälle

5.2 Stärken und Schwächen der Implementierung

6 Fazit

6.1 Entwicklungsstand

6.2 Ausblick auf zukünftige Entwicklungen

Abkürzungsverzeichnis

SOAP Simple Object Access Protocol

REST Representational State Transfer

AMQP Advanced Message Queuing Protocol

GRPC Google Remote Procedure Call

Literaturverzeichnis

[1] L. Wang, J. Tao, M. Kunze, A. C. Castellanos, D. Kramer, and W. Karl, "Scientific cloud computing: Early definition and experience." in *HPCC*, vol. 8, 2008, pp. 825–830.

[2] B. A. Nardi, S. Whittaker, and E. Bradner, "Interaction and outeraction: instant messaging in action," in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. ACM, 2000, pp. 79–88.

[3] M.-S. E. Scale, "Cloud computing and collaboration," *Library Hi Tech News*, vol. 26, no. 9, pp. 10–13, 2009.

[4] M. P. Papazoglou, "Service-oriented computing: Concepts, characteristics and directions," in *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*. IEEE, 2003, pp. 3–12.

[5] C. Pautasso, O. Zimmermann, and F. Leymann, "Restful web services vs. big'web services: making the right architectural decision," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 805–814.

[6] A. B. Hugo Haas. Web services glossary. [Online]. Available: https://www.w3.org/TR/ws-gloss

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 15. Juni 2016 Philipp Prögel