

Ergebnisbericht zum

DFG-Projekt



**Automatische Dienstvermittlung in
dienstorientierten Architekturen in Aktion
(ADD0action)**

Kurt Geihs, Steffen Bleul und Diana Comes

Universität Kassel



VERTEILTE SYSTEME

18. März 2010

Zusammenfassung

In diesem Bericht werden die Ergebnisse und Fortschritte des Forschungsprojekts ADDOaction vorgestellt. ADDOaction ist als Nachfolgeprojekt des Projekts ADDO entstanden und knüpft direkt an dessen Forschungsarbeiten an. Die Ergebnisse von ADDO wurden in ADDOaction ausgebaut und für noch offene Fragestellungen wurde eine Reihe von innovativen Lösungen erarbeitet.

Durch die Entwicklung in den letzten Jahrzehnten wurde das Internet zu einer wichtigen Infrastruktur für Geschäftsprozesse. Beliebige Anwendungen können als Dienste angeboten und übers Internet den Kunden online zur Verfügung gestellt werden. Eine flexible Dienstarchitektur ist dabei durch einen gewissen Grad an Dynamik gekennzeichnet, so dass Dienste angepasst, ausgetauscht oder entfernt werden können und eventuell gleichzeitig von verschiedenen Anbietern bereitgestellt werden können. Dienste müssen dabei sowohl die funktionalen als auch die nicht-funktionalen Quality of Service (QoS) Anforderungen der Klienten erfüllen, um Kundenzufriedenheit garantieren zu können. Die Vielzahl der angebotenen Dienste und die unterschiedlichen Anforderungen der Klienten machen eine manuelle Entdeckung und ein manuelles Management der Dienste praktisch unmöglich. ADDOaction adressiert genau diese Probleme einer dienstorientierten Architektur und liefert innovative Lösungen von der automatischen Entdeckung von Diensten bis hin zur Überwachung und zum Management von Diensten zur Laufzeit.

Forschungsgebiete, wie die automatische Dienstvermittlung und semantische Komposition von Diensten, die bereits Gegenstand des Vorgängerprojekts ADDO waren, wurden in ADDOaction weiter ausgebaut und vertieft. In unserem ADDOaction Framework werden Dienste automatisch über erweiterte semantische Dienstbeschreibungen (gemäß der neu entwickelten MECE Schnittstelle) und einem verbesserten Abgleichalgorithmus entdeckt. Diesbezügliche Ergebnisse aus ADDOaction wurden bereits in [15], [16], [11], [12], [9] veröffentlicht. Das Kompositionssystem für die semantische Komposition von Diensten wurde ebenfalls weiterentwickelt. Mit diesem Kompositionssystem haben wir auch im Jahr 2007 und im Jahr 2008 am internationalen Wettbewerb „Web Service Challenge“ teilgenommen und dabei den zweiten beziehungsweise dritten Platz erzielt. Die Ergebnisse und Erfahrungen vom Wettbewerb wurden in [13],[27], [29], [14] veröffentlicht.

Im Vergleich zu ADDO kamen in ADDOaction jedoch auch neue Forschungsthemen hinzu, wie beispielsweise die Überwachung und das Management von Geschäftsprozessen. Dies sind Aufgaben, die eine erfolgreiche dienstorientierte Architektur unbedingt gewährleisten muss. Damit Geschäftsprozesse ihre Aufgaben unter Einhaltung aller funktionalen und nicht-funktionalen Kriterien erfüllen, werden sie in unserem System zur Laufzeit überwacht. Im Falle unerwünschter Situationen wird sofort eingegriffen und fehlerhafte Dienste werden beispielsweise durch andere Dienste ersetzt. Hier kommt auch unsere neue Regel-basierte Sprache BPRules [20] zum Einsatz, die ein automatisches QoS-basiertes Management von Geschäftsprozessen unterstützt. Unser neuer QoS-Aggregationsalgorithmus [19] zeigt, wie die Laufzeitüberwachung der Dienste und die Berechnung der Werte von QoS-Eigenschaften des Geschäftsprozesses realisiert wird. Wir haben auch einen neuen Selektionsalgorithmus entwickelt, der Dienste entsprechend der Kundenanforderungen auswählt, um sie dann in den Geschäftsprozess einzubinden.

Im ADDOaction Framework wurden die entsprechenden Softwarekomponenten prototypisch implementiert, um unsere Ergebnisse praktisch demonstrieren und evaluieren zu können.

Inhaltsverzeichnis

Inhaltsverzeichnis	2
1 Arbeiten und Ergebnisse	3
1.1 Ausgangslage	3
1.2 Darstellung der erzielten Ergebnisse	4
1.2.1 Automatische Dienstvermittlung basierend auf semantischen Dienstbeschreibungen	4
1.2.2 Automatisches QoS Management von Diensten und Dienstprozessen	6
1.2.3 Semantische Komposition von Diensten: Der Wettbewerb „Web Services Challenge“	9
2 Ausblick	11
2.1 Ausblick auf künftige Arbeiten	11
2.2 Verwertungspotential	11
3 Publikationen	13
3.1 Zeitschriften und Tagungen	13
3.2 Technische Berichte	18
3.3 Preise und Auszeichnungen	19
4 Fazit	20
Literaturverzeichnis	21

1 Arbeiten und Ergebnisse

1.1 Ausgangslage

Das zentrale Forschungsgebiet von ADDOaction ist die automatische Dienstvermittlung und das Management von Diensten und Dienstkompositionen in dienstorientierten Architekturen. Dabei baut ADDOaction auf die erfolgreichen Ergebnisse des Vorgängerprojekts ADDO [10] auf und liefert Lösungen zu noch offenen Problemstellungen, die in ADDO noch nicht vollständig adressiert werden konnten.

Das Vorgängerprojekt ADDO verlief über zwei Jahre vom 15. Februar 2005 bis zum 31. März 2007. Dafür wurde eine wissenschaftliche Mitarbeiterstelle von der DFG finanziert. Weil ADDO sehr erfolgreich war und die Themengebiete in ADDO noch viel Potential für Forschung versprachen, wurde auch das Nachfolgeprojekt ADDOaction für noch weitere zwei Jahre genehmigt. Hierfür wurden für die Implementierung von ADDOaction zwei Mitarbeiterstellen von der DFG finanziert. Informationen über ADDO und ADDOaction haben wir auf unserer Webseite [1] veröffentlicht.

Zurzeit gewinnt das Internet bei der Abwicklung von Geschäftsprozessen immer mehr an Bedeutung und zeichnet sich als deren Infrastruktur aus. Dabei müssen Geschäftsprozesse den Anforderungen der Kunden gerecht werden und streng vertraulich sein. Die Standard-Technologie für die Realisierung von Geschäftsprozessen, die über das Internet abgewickelt werden, sind Web Services. Web Services bieten die Möglichkeit, eine Anwendung zu kapseln und deren Funktionalität als Dienst übers Internet den Kunden bereitzustellen.

Web Services basieren dabei auf offenen Internet Standards wie XML und HTTP und versprechen Interoperabilität in heterogenen Umgebungen, Wiederverwendbarkeit, Flexibilität und Reduzierung der Kosten. Über Web Services können Anwendungen unabhängig vom Anbieter oder von der gewählten Plattform problemlos in einen Geschäftsprozess integriert werden.

Weil Geschäftsprozesse über Organisationen hinweg laufen und die Dienste von verschiedenen Anbietern stammen können, besteht die Notwendigkeit, diese Dienste miteinander kommunizieren zu lassen. Web Services werden dabei zu einer sogenannten Komposition von Diensten zusammengestellt und realisieren damit einen komplexeren Geschäftsprozess (Service Process). Dienste können bei einem Partner oder Anbieter eingekauft werden und auf einfache Weise in den eigenen Geschäftsprozess eingebunden werden. Im Folgenden werden wir in diesem Bericht die Begriffe Geschäftsprozess, Dienstprozess, Service-Prozess und Business-Prozess als Synonym verwenden.

Die zahlreichen Dienstangebote werden von den entsprechenden Anbietern in einem Dienstvermittlungsverzeichnis (Service-Register) veröffentlicht. Um einen Dienst überhaupt nutzen zu können, muss er zunächst im Dienstvermittlungsverzeichnis gesichtet werden. Hierzu hat ADDO wichtige Ergebnisse für die automatische Dienstvermittlung geliefert. Die resultierenden Ansätze wurden in ADDOaction aufgegriffen und weiterentwickelt.

Dienste sind sowohl durch ihre funktionalen als auch durch ihre nicht-funktionalen Eigenschaften gekennzeichnet. Bei Web Services mit gleicher Funktionalität stellen die nicht-funktionalen Eigenschaften das entscheidende Kriterium dar, um zwischen den verschiedenen Angeboten zu differenzieren. Diese nicht-funktionalen Eigenschaften werden auch Quality of Service (QoS oder Dienstgüte) genannt. Beispiele für solche nicht-funktionale Eigenschaften sind Antwortzeit, Verfügbarkeit, Vertraulichkeit, Kosten oder die Durchsatzrate. Damit ein Dienst die Anforderungen seiner Kunden erfüllt, werden Verträge, sogenannte Service Level Agreements (SLAs), zwischen Dienstanbieter und Nutzer vereinbart. Um deren Einhaltung sicherzustellen, werden die Dienste und Dienstkompositionen während ihrer Laufzeit überwacht, und es wird überprüft, ob sich die QoS-Werte im gewünschten Rahmen befinden.

Vor diesem Hintergrund hat ADDOaction hauptsächlich folgende Ergebnisse und Fortschritte aufzuweisen:

1. Automatische Dienstvermittlung basierend auf semantischen Dienstbeschreibungen
 - Weiterentwicklung des *Abgleichalgorithmus* und Entwicklung der neuen semantischen Dienstbeschreibung *MECE- Mediation Contract Extension*
2. Semantische Komposition von Diensten
 - Weiterentwicklung des *Kompositionssystems* und Teilnahme am Wettbewerb „Web Services Challenge“ in den Jahren 2007, 2008 und 2009
3. Dynamische Einbindung und dynamischer Austausch von Diensten in Applikationen/ Geschäftsprozesse
4. Überwachung der QoS-Eigenschaften der Dienste und Berechnung der QoS-Werte für Geschäftsprozesse zur Laufzeit
 - Entwicklung des neuen *QoS-Aggregationsalgorithmus*
5. Überwachung der Einhaltung von SLA Verträgen für Dienste / Geschäftsprozesse
6. Automatisches QoS-basiertes Management von Diensten und Dienstprozessen
 - Entwicklung der neuen Regel-basierten Managementsprache *BPRules*
7. Auswahl von Diensten aufgrund von QoS-Eigenschaften und deren Integration in den Geschäftsprozess
 - Entwicklung eines neuen *Selektionsalgorithmuses*
8. Entwicklung eines Prototyps, anhand dessen die entwickelten Ansätze demonstriert und evaluiert wurden

Im Folgenden werden die Ergebnisse im Detail beschrieben und ein Einblick in die konkrete Realisierung der verschiedenen Ansätze gegeben. Durch die zitierten Referenzen wird gezeigt, auf welchen nationalen und internationalen Konferenzen und Workshops die Ergebnisse veröffentlicht wurden und wo weiterführende Details zu den Themen zu finden sind.

1.2 Darstellung der erzielten Ergebnisse

1.2.1 Automatische Dienstvermittlung basierend auf semantischen Dienstbeschreibungen

Publikationen: [15], [16], [11], [12], [9]

Ein wichtiges Kriterium für eine erfolgreiche dienstorientierte Architektur ist ihre Flexibilität. Dienste sollen entsprechend einer Dienstanfrage aus einem Dienstvermittlungsverzeichnis *automatisch entdeckt* werden können und anschließend auf einfache Weise in eine Applikation *eingebunden* werden können. Dienste können mit derselben Funktionalität aber zu unterschiedlichen QoS Levels und Preisen für die Kunden angeboten werden, so dass die Kunden eine Auswahlmöglichkeit vorfinden.

Falls Dienste, die in einer Applikation genutzt werden, die in den SLAs vereinbarten Anforderungen nicht mehr erfüllen, sollen diese durch neue Dienste mit den gewünschten Eigenschaften *automatisch ersetzt* werden können. Das Ersetzen von alten Diensten durch neue, zum Beispiel von einem anderen Anbieter, darf jedoch den normalen Ablauf der Applikation nicht stören. Für einen erfolgreichen Dienstersatz muss deshalb erst der gewünschte Dienst aufgesucht und entdeckt werden und kann erst anschließend in die Applikation integriert werden. Erschwerend kommt hierbei noch hinzu, dass eine Dienstschnittstelle eines Anbieters häufig unterschiedlich zu der Dienstschnittstelle eines anderen Anbieters ist.

In ADDOaction wurde ein Model entwickelt, welches eine automatische *Dienstvermittlung* und eine automatische *Integration und Ersetzung von Diensten* [15], [16], [11], [12] in einer Applikation/ Service-Prozess ermöglicht. Die *automatische Dienstvermittlung* wird von einem semantischen Service Broker unterstützt. Die zentrale Rolle des Service Brokers ist es, die Vermittlung zwischen dem Service Anbieter und dem Service Konsument zu ermöglichen. Dienstangebote vom Service Anbieter werden mittels des Service Brokers für die Klienten bereitgestellt. Der Service Broker ist eine zentrale Komponente im ADDOaction Framework und ist ähnlich wie ein UDDI [26] Service-Register aufgebaut. Der Service Broker ermöglicht, syntaktische und semantische Dienstbeschreibungen zu registrieren und zu deregistrieren und passende Dienste mit Hilfe eines entsprechenden Abgleichalgorithmuses zu entdecken. Details dazu können der Publikation [16] entnommen werden.

Eine *dynamische Integration und eine dynamische Ersetzung von Diensten* in einer Applikation / Geschäftsprozess sind erste Schritte, die für das Management von Diensten notwendig sind. In einer Umgebung mit einer großen Anzahl an Applikationen und Diensten ist eine manuelle *Integration oder Ersetzung* von Diensten in Service-Prozessen praktisch unmöglich. Deshalb ist es notwendig, eine Lösung zu finden, welche dynamisch zur Laufzeit eine automatische Integration oder Ersetzung eines Dienstes ermöglicht. Im ADDOaction Framework kommt dazu der sogenannte *Service Container* zum Einsatz, der als Service Proxy (eine Zwischenkomponente) fungiert (siehe [16]) und die Einbindung von Services in Applikationen unterstützt. Durch die Zusammenarbeit zwischen dem Service Broker und dem Service Container wird ein fehlerhafter oder unerwünschter Dienst durch ein besseres Dienstangebot aus dem Service Broker ersetzt.

In [15] stellen wir unseren ersten Ansatz für die semantische Entdeckung und Einbindung von Diensten in BPEL4WS [6] Service-Prozesse vor. Für einen Service-Prozess werden Dienste gesucht um sie danach im Service-Prozess automatisch einzubinden. Die semantische Entdeckung der Dienste für Service-Prozesse basiert auf einer *Service-Prozess Ontologie*. Die BPEL Dienstbeschreibung wird adaptiert so dass die Referenzen aus der BPEL Dienstbeschreibung auf die konkreten WSDL Referenzen (BPEL Imports, WSDL PortTypes, WSDL MessageParts) der entdeckten Dienste aktualisiert werden.

Die *Vermittlung von Diensten* basiert auf dem Verhältnis zwischen einer Dienstanfrage und Dienstangeboten. In der Dienstanfrage wird Auskunft über den gesuchten Dienst gegeben. Dies wird in der Web Service-Schnittstelle (Dienstbeschreibung) spezifiziert. Dienstanfragen und Dienstangebote sind in ADDOaction als semantische Erweiterung der OWL-S Ontologie dargestellt. Durch die Dienstvermittlung werden passende Dienstangebote für die Dienstanfrage ermittelt. Ein Dienstangebot muss sowohl syntaktisch als auch semantisch zur Dienstanfrage *passen*, um die Dienstanfrage erfolgreich erfüllen zu können. *Passen* bedeutet hierbei, dass es eine syntaktische und semantische Übereinstimmung zwischen Dienstanfrage und Dienstangebot gibt oder dass eine Anpassung womöglich über einen *Anpassungsprozess (Mediation)* geschaffen werden kann.

Im ADDOaction Framework wird der Anpassungsprozess durch einen neu entwickelten *Abgleichalgorithmus* realisiert. Er besteht aus zwei Teilen: dem funktionalen Abgleich und dem QoS-Abgleich. Ein funktionaler Abgleich von Diensten ist erfolgreich, wenn alle Operationen der Dienstanfrage im Dienstangebot vorhanden sind. Der Abgleich erfolgt in Abhängigkeit der Input/ Output Parametern der Service-Operationen und berücksichtigt deren semantische Darstellung. Hier werden zudem auch die Beziehungen (z. B. Spezialisierung) zwischen den semantischen Konzepten der Parameter in Betracht gezogen.

Unser Modell unterstützt auch die semantische Spezifikation von QoS-Anforderungen basierend auf einer Service Level Ontologie [16]. Der Abgleich für QoS-Anforderungen ist auf mehreren Ebenen möglich. Hier wird auch die Transformation von QoS-Metriken (z.B: Sekunden, Millisekunden) und eine Transformation von Prädikaten (z. B. größer, kleiner, etc.) und QoS-Dimensionen (z. B. Durchsatz) unterstützt.

Das *Message Mediation System*, das für die Entdeckung der Dienste und die Anpas-

sung von Dienstmeldungen verantwortlich ist, wird ausführlich in [11] beschrieben. Um die gewünschte Dienstschnittstelle zu erhalten, passt der Algorithmus die erwarteten Anfrage- bzw. Antwortmeldungen des Klienten bzw. des Diensteanbieters durch entsprechende Transformationen an.

Für den Abgleich und die Anpassung von Dienstfragen und -angeboten definieren wir eine semantische Erweiterung der Dienstschnittstellenbeschreibung in WSDL [17] (Web Services Description Language), die wir *MECE- Mediation Contract Extension* nennen. Über unsere neue MECE Lösung werden die syntaktischen WSDL Elemente semantisch annotiert, wobei jedoch das WSDL Dokument immer noch standardkonform bleibt. Weitere Details über die MECE Dienstschnittstellenbeschreibungen können unseren Publikationen [9], [12] entnommen werden. Für semantisch äquivalente, aber syntaktisch unterschiedliche Meldungen werden XSLT [18] Transformationen zur Überbrückung der Differenzen automatisch generiert. In unserem System können so auch im Falle von unterschiedlichen strukturellen Repräsentationen, Datentypen und Masseinheiten Dienstangebote genutzt werden.

Der gesamte Abgleich- und Anpassungsprozess bleibt dabei sowohl für den Diensteanbieter als auch für den Klienten vollkommen transparent. Um Konvertierungsroutinen von Datentypen und Messeinheiten bereitzustellen, können Plug-ins im System registriert werden. Plug-ins sind Software Komponenten (z. B. Java Klassen), die entsprechende Transformationen (z. B. eine Währungsumrechnung von Euro nach Dollar) unterstützen und sowohl vom Diensteanbieter als auch vom Klienten bereitgestellt werden können. Differenzen zwischen unterschiedlichen Meldungsformaten werden somit durch automatisch generierte XSLT Dokumente und die bereitgestellten Plug-ins überbrückt.

Das Ergebnis des Abgleichprozesses wird durch den sogenannten *Auswertungsrang* (ranking score) repräsentiert, welcher ein Maß für den Erfolg des Abgleichprozesses darstellt. Der Auswertungsrang wird in Abhängigkeit von der semantischen Distanz der abgeglichenen semantischen Konzepte und der Anzahl der benötigten Plug-ins berechnet. Das Message Mediation System liefert für eine Dienstfrage eine geordnete Liste von Dienstangeboten, welche gemäß dem Auswertungsrang sortiert ist. Bei misslungenen Abgleichungen werden auch ausführliche Berichte ausgegeben, so dass potentielle Probleme schneller entdeckt und behoben werden können.

Durch unser Mediation System können aber auch semantisch äquivalente Dienste dynamisch eingebunden (oder ersetzt) und in einen Service-Prozess integriert werden (siehe [9], [12]).

Um die Vorteile unseres Ansatzes zu demonstrieren, wurde eine Applikation prototypisch realisiert. Unser Beispielszenario (siehe [9],[12]) besteht aus einem *Bookshop* Geschäftsprozess, der in BPEL4WS implementiert wurde und bei dessen Realisierung mehrere Dienste im Prozess eingebunden sind. Die für den Prozess am besten passenden/geeigneten Dienste wurden dabei über das Mediation System mit Hilfe des oben beschriebenen Abgleichprozesses bestimmt und anschließend in den Service-Prozess automatisch eingebunden.

1.2.2 Automatisches QoS Management von Diensten und Dienstprozessen

Publikationen: [8], [9], [20], [19]

Ein wichtiges Forschungsgebiet von ADDOaction war das Management von Service-Prozessen im Hinblick auf QoS-Eigenschaften. Das Managementverfahren sorgt für eine möglichst einwandfreie und fehlerfreie Abwicklung von Geschäftsprozessen zur vollen Kundenzufriedenheit.

Bei der Abwicklung eines Service-Prozesses (Geschäftsprozesses) in einer dienstorientierten Architektur nehmen meist mehrere Services von unterschiedlichen Partnern oder Organisationseinheiten teil. Diese Interaktion von lose gekoppelten Diensten bei der Abwicklung eines Service-Prozesses wird als Dienstkomposition realisiert. Ein Service-Prozess ist gleichzeitig wiederum ein Dienst und kann als Web Service zur Verfügung gestellt werden. Beispiele

von Geschäftsprozessen (als Dienstkompositionen realisiert) sind das Kaufen von Produkten in einem Online Shop, die Reservierung von Fahrkarten bei einer Fluggesellschaft oder die Anfrage von einem Kredit bei einer Bank.

Zur Beschreibung der Komposition sind mehrere Sprachen bekannt wie beispielsweise die Web Services Flow Language (WSFL) von IBM, die Web Services Conversation Language (WSCL) von W3C oder XLANG von Microsoft. Jedoch hat sich für die Beschreibung und Ausführung von Business Prozessen die Web Services Business Process Execution Language WS-BPEL [21] als Standard etabliert.

Um sicherzustellen, dass die Dienste/Geschäftsprozesse die Anforderungen der Kunden erfüllen, werden *Verträge (Service Level Agreements SLA)* zwischen Dienstanbieter und Nutzer geschlossen. Im Vertrag wird vereinbart, welche Qualitätsmerkmale der Dienst einzuhalten hat. In unserem Framework werden Dienste zur Laufzeit überwacht und anhand des Vertrags wird überprüft, ob sich die QoS-Messwerte noch im vereinbarten Rahmen befinden oder nicht. Im Falle von Vertragsverletzungen können verschiedene Aktionen in Kraft treten, wie zum Beispiel eine Benachrichtigung oder eine Kostenreduzierung.

Zur Laufzeit können jedoch auch unerwünschte Situationen auftreten, zum Beispiel der Ausfall eines Web Services, welche ein unerwünschtes Verhalten des gesamten Prozessablaufs mit sich bringen. Weitere Beispiele von möglichen Problemen sind das Auftreten eines Netzwerkfehlers, ein Dienst antwortet nicht in der gewünschten Zeit, ist nicht mehr verfügbar oder erfüllt nicht mehr die Anforderungen.

Ein wichtiges Ziel von ADDOaction war es, dass auch bei solch unerwünschten Situationen der gesamte Prozess so gesteuert und verwaltet wird, dass der Prozessablauf möglichst nur minimal gestört wird. Im Falle von Störungen sollen verschiedene Aktionen zur Verfügung stehen, die es erlauben, das QoS-Verhalten des Prozesses zu steuern und zu korrigieren. Zum Beispiel sollen Web Services mit schlechten QoS-Eigenschaften automatisch zur Laufzeit durch andere Services mit besseren QoS-Eigenschaften ersetzt werden. Für die Lösung des Problems haben wir das *Managing Services with QoS Constraints (MASQ) System* als Prototyp entwickelt. Hierbei wurden innovative Strategien und Algorithmen entwickelt, die das QoS Management unterstützen.

Um einen Service-Prozess erfolgreich zu verwalten, ist zunächst sein Verhalten hinsichtlich der vereinbarten QoS-Eigenschaften zu untersuchen. Hierzu muss der Geschäftsprozess während der Laufzeit überwacht und die Dienstgüte Eigenschaften aufgezeichnet werden. Verhält sich der Prozess nicht so wie erwartet, so müssen entsprechende Management- Entscheidungen und Maßnahmen zur Verbesserung der Dienstgüte getroffen werden. Diese Aufgaben wurden als verschiedene Software Komponenten realisiert und in unser MASQ System integriert. MASQ ist ein *self-managed* System, da es gleichzeitig auch dem MAPE (Monitor, Analyse, Plan, Execute) Konzept [24] folgt.

Im Folgenden wird ein Überblick der erzielten Ergebnisse hinsichtlich des QoS Managements dargestellt:

1. Design und Entwicklung des *MASQ Systems* als self-managed System für Service-Prozesse
2. Entwicklung von Geschäftsprozessen (zum Beispiel: Bookshop) zur Demonstration der Ansätze und Evaluierung
3. Überwachung und Messen von QoS-Parametern von Geschäftsprozessen und einzelnen Web Services (*die QoS-Monitor und Aggregator Komponente*). Hierzu wurde auch der QoS-Aggregationsalgorithmus entwickelt.
4. Konzeption und Entwicklung der neuen *BPRules* Sprache für das QoS-basierte Management von Business Prozessen (*die BPRules Manager und Evaluator Komponente*)
5. Automatische Steuerung und automatisches Management von Geschäftsprozessen (*die Process Management Komponente*)

6. Entwicklung des neuen *Selektionsalgorithmuses* für die Auswahl von Web Services aus dem Service-Register unter Berücksichtigung der QoS-Eigenschaften.

Der *Bookshop* Geschäftsprozess wurde entwickelt, um das QoS-basierte Management von Prozessen zu veranschaulichen und dadurch auch die Realisierbarkeit unserer Ansätze zu demonstrieren. Der Prozess ist als eine Web Service Komposition definiert und dient dem Kauf von Büchern in einem Online Shop. Für die Implementierung wurde BPEL4WS ausgewählt. Bei der Entwicklung wurde darauf geachtet, dass der Geschäftsprozess möglichst einer realistischen Anwendung in einem Unternehmen entspricht. Im Prozess wurden vier Web Services eingebunden: ein *Lager Service*, ein *Großhändler*, ein *Büchershop* und ein *Bank Service*. Der Prozess startet mit dem Ankommen einer Buchnachfrage eines Kunden. Daraufhin wird vom Lager Service geprüft, ob sich die Bücher im Lager vorrätig sind. Falls dies nicht der Fall ist, werden die Bücher beim Großhändler mit Hilfe des entsprechenden Services eingekauft. Die Rechnung für den Kunden wird vom Büchershop Service erstellt. Anschließend wird der Bank Service aufgerufen, um die Geldüberweisung vom Kundenkonto auf das Büchershop Konto durchzuführen. Ein anderes Beispielszenario (siehe [20]), welches noch untersucht wurde, war der *DrivingLicence* Prozess für die Beantragung eines Führerscheins beim entsprechendem Verwaltungsamt.

Der komplette Service-Prozess wird hinsichtlich seiner QoS-Eigenschaften überwacht. Die QoS-Werte der einzelnen Web Services tragen zu den QoS-Werten der gesamten Service Komposition bei. QoS kann von verschiedenen Parteien gemessen werden, beispielsweise vom Service Anbieter selbst oder aber auch vom Klienten. Die QoS-Daten können dementsprechend in verschiedenen verteilten Ressourcen gespeichert werden, was im Überwachungsprozess berücksichtigt werden muss. Aus diesem Grund haben wir einen flexiblen Ansatz zum Messen und Aggregieren von QoS-Eigenschaften für Service-Prozesse realisiert. Neue QoS-Parameter können leicht dem System hinzugefügt werden und es wird nur minimaler manueller administrativer Aufwand benötigt. Dies wird auch durch das automatische Sensor Deployment möglich, das für das Messen der QoS-Eigenschaften zuständig ist. Der Ansatz wurde mittels einer prototypischen Implementierung durch die *QoS Monitor und Aggregator Komponente* veranschaulicht. Weiterhin haben wir einen innovativen generischen Algorithmus, der *QoS-Aggregationsalgorithmus*, zur Aggregation der QoS Eigenschaften in Web Service Kompositionen entwickelt. QoS-Parameter wie Kosten, Antwortzeit, Verfügbarkeit und Zuverlässigkeit(reliability) werden gemessen und zusammengerechnet. Weitere Details zu diesem Thema haben wir in [19] veröffentlicht. Der QoS-Aggregationsalgorithmus [19] stellt eine Methode zur QoS-Aggregation für einen Pfad (eine Ausführung des Prozesses) dar. Falls der Ausführungspfad nicht vorliegt, werden stattdessen die Wahrscheinlichkeiten der Ausführungspfade berücksichtigt. Die QoS-Eigenschaften des Prozesses werden nach demselben Prinzip estimiert.

Nachdem die QoS Eigenschaften der atomaren Services und des gesamten Prozesses inspiziert wurden, müssen entsprechende Entscheidungen für das Prozessmanagement getroffen werden. Hierzu haben wir eine neue Sprache, *BPRules* [20] für das QoS Management von Geschäftsprozessen entwickelt. BPRules ist eine regelbasierte Sprache, die eine innovative Lösung für das Service Management bietet. Die Regeln werden in sogenannten BPR-Dokumenten in der XML Syntax spezifiziert und vom BPRules XSD Schema validiert. Eine Regel besteht aus einer QoS-Bedingung und einer Aktion, wobei die Aktion im Falle der Erfüllung der QoS-Bedingung aufgerufen wird. Dadurch kann über die QoS-Bedingung überprüft werden, ob die QoS Eigenschaften im gewünschten Wertebereich liegen oder nicht. Zusätzlich können aber auch Regeln ganz auf die individuellen Wünsche der Kunden zugeschnitten werden. BPRules bietet eine Reihe an nützlichen Aktionen, die zur Korrektur des QoS Verhaltens des Prozesses einsetzbar sind. Weitere Details über BPRules wurden bereits in [20] veröffentlicht. Folgende wichtige Vorteile von BPRules sind hierbei zu nennen: Spezifikation und Verwaltung des gesamten Prozesses als auch von einzelnen Sektionen, Anwendung von Regeln auf einer Menge von Prozessinstanzen und das Ändern und Aktualisieren von Regeln

dynamisch zur Laufzeit.

Über Sektionen können nur Teile des Prozesses angesprochen werden und QoS-Bedingungen können auf einzelnen Sektionen beschränkt werden. Dies ermöglicht eine bessere Sicht auf den Ablauf des Prozesses und es wird schneller ersichtlich, welche Sektionen als Ursache für unerwünschte Situationen verantwortlich sind. Unser Ansatz unterstützt auch, Beziehungen zwischen QoS-Parametern aus unterschiedlichen Sektionen zu definieren. Beispielsweise soll die Antwortzeit in einer Sektion nur ein Drittel der Antwortzeit des gesamten Prozesses betragen. Regeln können auf einer Untermenge von Prozess Instanzen (Ausführungen) angewandt werden, wie zum Beispiel die laufenden oder abgeschlossenen Instanzen, oder auf 30 Prozent der Instanzen.

Durch korrektive Aktionen wird auf den Prozessablauf Einfluss genommen. Wir unterscheiden zwischen *Management Aktionen* und *allgemeinen Aktionen*. Als Management Aktionen gelten beispielsweise das Deployen, Undeploymen, Starten, Stoppen des Prozesses, Ersetzen von Web Services innerhalb des Prozesses, Melden von Fehlern oder Benachrichtigen von Interessenten über bestimmte Ereignisse. Im BPR-Dokument können mehrere Regeln zu den sogenannten *Rulesets* gruppiert werden. Darauf können die allgemeinen Aktionen angewandt werden, sowie das Aktivieren, Deaktivieren oder Aktualisieren von Rulesets.

Zur Demonstration des QoS Managements von Prozessen mittels BPRules wurde eine prototypische Implementierung durch die *BPRules Manager und Evaluator Komponente* realisiert. Diese Komponente ist für die Evaluierung der BPR-Dokumente zuständig und delegiert die entsprechenden Aktionen an die *Process Management* Komponente. Diese Komponente ist wiederum für die Steuerung und die Ausführung der korrektiven Aktionen zuständig.

Mit Hilfe des *Selektionsalgorithmus* werden Dienste mit der gewünschten Funktionalität und QoS im Service-Register aufgesucht und schließlich in den Prozess eingebunden. Im Service-Register ist mit vielen Services mit gleicher Funktionalität und unterschiedlicher QoS zu rechnen. Stehen für mehrere Services viele Alternativen zur Verfügung, wächst die Anzahl der möglichen Kompositionen exponentiell in der Anzahl der im Prozess benötigten Services und das resultierende Optimierungsproblem ist NP hart. Für die Lösung des Problems werden ein genetischer Algorithmus aber auch neue Ansätze wie unser Optim-S Algorithmus und ein Gradientenverfahren untersucht. Wir beabsichtigen die drei Verfahren miteinander zu vergleichen und die Ergebnisse zu veröffentlichen. Unser Optim-S Algorithmus hat den Vorteil, brauchbare Lösungen zu finden und das Suchverfahren dynamisch zur Laufzeit anzupassen. Während in einigen Fällen (zum Beispiel in mobilen Anwendungen) eine brauchbare Lösung in kürzester Zeit benötigt wird, kann in anderen Fällen (zum Beispiel zur Entwicklungszeit) mehr Zeit zur Findung der optimalen Lösung investiert werden. Mit dem Optim-S Algorithmus kann das Verhältnis zwischen Komplexität des Algorithmus und Optimalität der Lösung in Abhängigkeit von den jeweiligen Anforderungen ganz einfach geregelt werden. Ein anderer Ansatz, den wir für das Selektionsproblem untersuchen, ist ein Gradientenverfahren, welches unter gewissen Umständen eine Lösung nahe dem Optimum oder sogar die optimale Lösung verspricht. Uns ist nicht bekannt, dass dieses Verfahren bislang für die Serviceauswahl eingesetzt wurde.

1.2.3 Semantische Komposition von Diensten: Der Wettbewerb „Web Services Challenge“

Publikationen: [13],[27], [29], [14], [23], [7]

Der Wettbewerb findet im Rahmen der IEEE Joint Conference on E-Commerce Technology (CEC) und Enterprise Computing, E-Commerce und E-Services (EEE) ([4], [5]) jährlich statt. Er bietet internationalen Teams die Möglichkeit, ihre Forschungsergebnisse vorzustellen und miteinander zu vergleichen.

Im Jahr 2006 erreichte unser Team den ersten Platz, im Jahr 2007 den zweiten Platz und im Jahr 2008 wurde unser Kompositionssystem auf Platz drei gewertet. Unser Komposi-

tionssystem entdeckt passende Web Services, die kombiniert werden, um eine Dienstanfrage zu erfüllen. Dies geschieht aufgrund der semantischen Annotationen der Input- und Output-Parameter der Dienste, die an der Komposition beteiligt sind. Um eine Dienstanfrage erfolgreich zu erfüllen, muss die Ausführung der Dienstkomposition alle gewünschten Output-Parameter der Dienstanfrage liefern. In [27] haben wir für die semantische Dienstkomposition drei verschiedene Ansätze beschrieben und miteinander verglichen: ein IDDFS-, Greedy- und einen genetischen Ansatz. Unsere Ergebnisse haben wir in [13], [27], [29] veröffentlicht.

Aufgrund der Erfolge in den Vorjahren 2008 und 2009 wurde unser Team auch in der Organisation und in das Management des Wettbewerbs involviert. In 2008 haben acht internationale Teams von verschiedenen Universitäten weltweit, unter anderem aus den U.S.A, China, Niederlande, etc. teilgenommen. Wir konnten mit Freude feststellen, dass dieses Themengebiet Begeisterung bei den Forschungsteams in aller Welt weckt und dass immer mehr Teams Interesse zeigen, am diesem Wettbewerb teilzunehmen. Durch die Annäherung an realistische Szenarien haben wir dazu beigetragen, dass die WS-Challenge eine noch praxisorientiertere Veranstaltung wurde. Aus diesem Grund haben wir neue Regeln eingeführt und der Wettbewerb 2008 basierte bereits auf standardisierten Datenformaten wie OWL [25], WSDL [17] und WSBPEL [21]. Wir haben auch einen neuen Testset-Generator entwickelt, der Konfigurationen wie in einer industrienahen dienstorientierten Architektur produziert. Der Wettbewerb hatte somit noch komplexere Herausforderungen an seine Teilnehmer gestellt. Unsere Erfahrungen diesbezüglich haben wir auch in [14] veröffentlicht.

2 Ausblick

2.1 Ausblick auf künftige Arbeiten

Mit der Ausbreitung des mobilen Internets werden Web Services immer mehr auch in mobile Applikationen integriert. Im Bereich von **Ubiquitous Computing** steht der Benutzer im Mittelpunkt, der von der Technik überall und allgegenwärtig unterstützt wird. In einer derartigen Umgebung ist man aber meist mit Diensten und Anwendungen von verschiedenen Anbietern konfrontiert, die auf unterschiedlichen Plattformen laufen. Wie wir wissen, unterstützen Webservices die Interoperabilität zwischen heterogenen Anwendungen in verteilten Umgebungen. Deswegen stellen Webservices auch eine sehr gute und realistische Lösung für das Zusammenspiel von Diensten in einer ubiquitären Umgebung dar.

Ubiquitous Computing (UC) ist auch von Dynamik und Unsicherheit gekennzeichnet, denn überall in der Umgebung werden neue Dienste dem Benutzer angeboten, während andere Dienste wieder verschwinden. Eine mobile Applikation soll für den Benutzer von Nutzen sein, dabei aber wenig Aufwand und Aufmerksamkeit des Benutzers erfordern, damit er nicht von seinen Tätigkeiten abgelenkt wird. Aus diesem Grund soll die mobile Applikation automatisch neue Dienste einbinden/oder entfernen können und eine spontane Interaktion zwischen den ubiquitären Diensten ermöglichen. ADDOaction hat bereits wichtige Ergebnisse für die automatische Entdeckung, Integration, Ersetzung und Komposition von Webservices und deren Selbstmanagement erzielt, die auf funktionale (semantische) und QoS-Anforderungen basieren. In einer ubiquitären Anwendung soll jedoch gleichzeitig auch Rücksicht auf die Präferenzen und Intentionen des Nutzers der Applikation genommen werden. Während wir uns in ADDOaction auf Geschäftsprozesse, QoS und funktionale Anforderungen konzentriert haben, treten im Ubiquitous Computing auch andere spezielle Einschränkungen hinzu, wie beispielsweise beschränkte Kapazitäten von Endgeräten, die Präferenzen des Benutzers oder die Umgebung in der er sich befindet.

Eine UC Applikation muss sich auch an die Situative Umgebung des Benutzers adaptieren, zum Beispiel wenn sich der Benutzer in einem Einkaufszentrum befindet und ihm verschiedene Dienste, von mehreren Läden, angeboten werden. Auch in diesem Fall soll der bestmögliche Dienst für die jeweilige Situation in dem sich der Benutzer befindet ausgewählt werden. Dabei muss aber auch auf die rechtlichen Rahmenbedingungen (zum Beispiel Datenschutz) und Vertraulichkeit Rücksicht genommen werden, denn nur so ist eine erfolgreiche Interaktion zwischen den Diensten möglich. Man kann hier also neue Herausforderungen auch bei der Komposition und im Zusammenspiel von Diensten erkennen. UC Applikationen sind durch einen hohen Grad an Automatisierbarkeit gekennzeichnet. Dadurch dass ADDOaction ein Framework für eine automatisierte Interaktion zwischen Diensten liefert, können die Ergebnisse von ADDOaction weiterverwendet werden und auch detaillierter in Richtung von Ubiquitous Computing weiterentwickelt werden. Das wird unter anderem ein Thema des neuen interdisziplinären Forschungsschwerpunkts VENUS [3] an der Universität Kassel sein.

2.2 Verwertungspotential

Die Ergebnisse aus ADDOaction können sowohl in industriellen Projekten als auch in wissenschaftlichen Forschungsprojekten weiterverwendet werden. Bisher wurden schon einige Ergebnisse aus ADDOaction in das EU Forschungsprojekt MUSIC [2] eingebracht. Der Ansatz von MUSIC [22] zur semantischen Modellierung von Service Levels bei der Entdeckung von QoS basierten Diensten ist deshalb nahezu identisch mit den in ADDOaction entwickelten Konzepten. Die gemeinsame Basis für beide Forschungsprojekte stellt dabei die QoS basierte Dienstbeschreibung mit Hilfe von OWL-S und die semantische Modellierung von QoS Dimensionen dar.

Die Ziele aus ADDOaction wurden auch dementsprechend festgelegt, so dass auch der wirtschaftliche Nutzen Beachtung findet. Da in einem erfolgreichen Geschäftsprozess die Kun-

denzufriedenheit im Mittelpunkt steht, haben wir mit unserem Ansatz zum QoS Management von Geschäftsprozessen wichtige Beiträge für einen ungestörten Prozessablauf geliefert. Die starke Verbreitung der Web Service Technologie und die Abwicklung von Geschäftsprozessen über das Internet erfordert automatisierte Lösungen für die Dienstvermittlung und das Management von Diensten. Dieser Bedarf wurde in ADDOaction erkannt und die automatisierten Lösungen aus ADDOaction sind für industrielle Projekte von Nutzen. Die Software Komponenten aus ADDOaction bieten Schnittstellen an, so dass sie in anderen Applikationen auch leicht zu integrieren sind.

3 Publikationen

3.1 Zeitschriften und Tagungen

1. Flexible Automatic Service Brokering for SOAs [16]

Steffen Bleul, Michael Zapf, Kurt Geihs

Proceedings on 10 th IFIP / IEEE Symposium on Integrated Management (IM 2007), Munich, Germany, Mai, 2007

Abstrakt: *Enterprises apply SOAs to implement business processes as services which are adaptive to software changes and updates. Service management has to ensure high availability, performance and fault tolerance and is therefore crucial for the undisturbed functionality of a SOA. However, manual service management is cumbersome with a large amount of services and applications. Services that fail or do not achieve a certain Quality of Service (QoS) need to be manually replaced. Up to now, a fully automatic approach is missing. We introduce a semantic model and architecture for automatic service brokering in Service-Oriented Architectures (SOA). A Service Broker is used for semantic service discovery with respect to functional criteria and QoS. Hence, we use a semantic matching algorithm on an extension of the OWL-S ontology. Services are requested by Service Containers which monitor and replace services in interaction with the Service Broker. The architecture is presented with an implementation using Web Services.*

2. Automatic Service Process Administration by Semantic Service Discovery [15]

Steffen Bleul, Michael Zapf, Kurt Geihs

Proceedings on 7th International Conference on New Technologies of Distributed Systems (NOTERE 2007), Marrakech, Maroc, Juni, 2007

Abstrakt: *One of the salient features of Service-Oriented Architectures is that services can be deployed and removed at runtime. Service replacement and management for service processes is a demanding task in complex environments, especially under additional constraints like minimizing downtime. We introduce an approach for automatic service process administration. A service process ontology allows to describe service processes with roles and the functionality of the required roles. Semantic annotations allow to compare services with compatible behavior but different operation, parameter, or task naming. The matching result is specified with our service result ontology. We describe a suitable procedure to automatically modify service execution descriptions, which can be utilized for achieving self-manageability of service processes.*

3. Quality-aware Self-Management for Service Processes in Service-Oriented Architectures [8]

Steffen Bleul

Proceedings of the IBM PhD Student Symposium at ICSOC 2007, Vienna, Austria, 2007

Abstrakt: *One of the salient features of Service-Oriented Architectures is that services can be deployed and removed at runtime. But service replacement and management for service processes is a demanding task in complex IT-Systems, especially under additional constraints like optimizing the Quality of Service of a service process. A self-managing system is desired but missing. We have already achieved self-healing and self-optimization with our service brokering system. We apply ontologies to discover*

service alternatives and their QoS. In this paper we not only present our work on quality-aware service discovery but also propose a self-manageable infrastructure for service processes. The infrastructure can be dynamically instantiated, configured and bound to management endpoints with semantic service discovery. The approach not only automates the binding of service management systems of multiple vendors but also the SLA monitoring and the ad-hoc integration of services in service processes.

4. Self-Integration of Web Services in BPEL Processes [9]

Steffen Bleul, Diana Comes, Marc Kirchhoff, Michael Zapf
Proceedings of the Workshop Selbstorganisierende, Adaptive, Kontextsensitive verteilte Systeme (SAKS), Wiesbaden, Germany, 2008

Abstrakt: *Interoperability between clients and service may not be seen as a major challenge in SOAs but in reality services under change impose a major hindrance in service management. We present a model and system for service process management where we achieve self-integration by automatic message matching and runtime transformation. We have developed the necessary WSDL schema extension, a semantic discovery algorithm, and a runtime mediation system. Our matching algorithm can detect semantically related message elements and generate appropriate XSLT documents. Finally our system dynamically instantiates mediators to bind services to service processes specified with BPEL4WS.*

5. Different Approaches to Semantic Web Service Composition [27]

Thomas Weise, Steffen Bleul, Diana Comes, Kurt Geihs
Proceedings of the 2008 Third International Conference on Internet and Web Applications and Services, IEEE Computer Society, Athens, 2008

Abstrakt: *Semantic web service composition is about finding services from a repository that are able to accomplish a specified task if executed. The task is defined in a form of a composition request which contains a set of available input parameters and a set of wanted output parameters. Instead of the parameter values, concepts from an ontology describing their semantics are passed to the composition engine. The parameters of the services in the repository the composer works on are semantically annotated in the same way as the parameters in the request. The composer then finds a sequence of services, called a composition. If the input parameters given in the request are provided, the services of this sequence can subsequently be executed and will finally produce the wanted output parameters. In this paper, three different approaches to semantic web service composition are formally defined and compared with each other: an uninformed search in form of an IDDFS algorithm, a greedy informed search based on heuristic functions, and a multiobjective genetic algorithm.*

6. Automated Management of Dynamic Web Service Integration [11]

Steffen Bleul, Kurt Geihs, Diana Comes, Marc Kirchhoff
Proceedings of the 15th Annual Workshop of HP Software University Association (HP-SUA). Infocomics-Consulting, Stuttgart, Germany, 2008

Abstrakt: *A Service Oriented Architecture promises a high potential for service integration. Business applications built with the SOA paradigm should be able to change and integrate additional business logic easily. However, replacing a service and integrating another service often needs to cope with interface mismatches. This requires transforming exchanged messages, and in general an efficient support*

for integration management. In this paper we introduce an automated approach for the management of service integration. It is based on a new Mediation Contract Extension (MECE) that allows semantic annotation for XSD type definition inside WSDL documents. Our discovery algorithm deduces automatically XSLT stylesheets in order to allow mediation between heterogeneous Web Service interfaces. This used in a distributed management framework to manage dynamic ad-hoc service integration with semantically related but syntactically different messages. The discovery engine along with XSLT stylesheet creation enables autonomous dynamic self-reconfiguration of the whole system.

7. Management of Business Processes with the BPRules Language in Service Oriented Computing [20]

Diana Comes, Steffen Bleul, Michael Zapf
Workshops der 16. ITG/ GI Fachtagung: Kommunikation in Verteilten Systemen (WowKiVS), Kassel, Germany, 2009

Abstrakt: *Quality of Service (QoS) concerns are an important topic for the realization of business processes. While BPEL is considered the de facto standard for web service compositions, QoS requirements are not part of its specification. We present the BPRules (Business Process Rules) language for the management of business processes with respect to QoS concerns. BPRules is a rule-based, declarative language which brings novel benefits in the management of business processes, like QoS dependability for sub-orchestrations and corrective actions tailored to the specific needs of the clients. We present the main constructs of the BPRules language and how they support the flexible adaptation of the business process during runtime. Decision making is done according to the behavior of several process executions. An illustrative scenario shows how BPRules is applied to a business process.*

8. Automated Integration of Web Services in BPEL4WS Processes [12]

Steffen Bleul, Kurt Geihs, Diana Comes, Marc Kirchhoff
16. ITG/ GI Fachtagung: Kommunikation in Verteilten Systemen (KiVS), Kassel, Germany, 2009

Abstrakt: *In order to fully exploit the potential of dynamic service-oriented architectures based on Web Services we provide a novel self-integration service infrastructure that supports automatic service discovery and reconfiguration for BPEL4WS processes. Our service discovery approach takes into account possible runtime transformations such that a service in a BPEL4WS process can be replaced by a semantically similar service even if service interfaces and message structures do not match. We present the main building blocks of our solution, i.e. a standard-conformant WSDL schema extension called Mediation Contract Extension (MECE), a corresponding semantic discovery algorithm, and a runtime mediation system that generates appropriate XSLT transformations on the fly. Our solution dynamically instantiates mediators to bind services to service processes specified with BPEL4WS.*

9. A Flexible Approach for Business Processes Monitoring [19]

Diana Comes, Steffen Bleul, Thomas Weise, Kurt Geihs
Proceedings of the 9th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS 2009), Lissabon, Portugal, 2009

Abstrakt: *Business processes and their implementation as Web Service Compositions are not only dependent on Web Services and partners all over the Internet, but also on their failsafe execution. Service providers have to obligate their services to perform according to negotiated Quality of Service (QoS) parameters. For example, response time and throughput are important parameters to achieve fast and efficient services. Overloaded or failing services may compromise the reliability and execution of whole*

enterprise processes. In this paper we introduce a flexible monitoring approach for the measurement of QoS in BPEL (Business Process Execution Language) processes. We propose a generic algorithm for QoS aggregation in BPEL processes. The novel generic aggregation algorithm applies customized aggregation functions for QoS dimensions. Furthermore, we present a BPEL monitoring system which supports ad-hoc sensor deployment and efficient runtime and offline data aggregation not only

10. PIK-Praxis der Informationsverarbeitung und Kommunikation

Nicolas Repp, Sebastian Hudert, Steffen Bleul

PIK - Praxis der Informationsverarbeitung und Kommunikation. Band 31, Heft 4, 2008

Abstrakt: *The application of the SOC paradigm to real world scenarios provides research and business challenges both in the present and the future of the distributed systems as well as the software and enterprise engineering disciplines. After reviewing the submitted papers, we are sure that the selected papers indicate a distinctive research focus on the development and provisioning of high-quality services and systems.*

11. Making a Fast Semantic Service Composition System Faster [13]

Steffen Bleul, Thomas Weise, Kurt Geihs

Proceedings of the 2007 IEEE Joint Conference on E-Commerce Technology (CEC'07) and Enterprise Computing, E-Commerce and E-Services (EEE'07), Tokyo, Japan, 2007

Abstrakt: *Service Oriented Architecture (SOA) is a flexible software design paradigm for enterprises. The workflows of a company are implemented as services which can be arranged, updated and managed at runtime without interfering with ongoing business. Service management aims at providing undisturbed access to services. Its efficiency strongly depends on a fast response time in the case of a failure. This is hard to achieve since the relations between applications and services require comprehensive knowledge and lack transparency for administrators. Self-organizing approaches promise a solution by automating service discovery. In this paper we present a service discovery system which enables service compositions from semantic descriptions stored in a knowledge base. Therefore, it utilizes multiple composition algorithms from which the most appropriate set is selected and applied according to the size of the knowledge base and the available processors. The functionality of our system is made available through a Web Service interface itself. It is thus applicable in self-organizing service management systems with any number of services and ontologies.*

12. Semantic Web Service Composition for Service-Oriented Architectures [29]

Thomas Weise, Steffen Bleul, Marc Kirchhoff, Kurt Geihs

Proceedings of IEEE Joint Conference (CEC/EEE 2008) on E-Commerce Technology (Tenth CEC'07) and Enterprise Computing, E-Commerce and E-Services (Fifth EEE'08), 2008

Abstrakt: *Semantic web service composition is about finding services from a repository that are able to accomplish a specified task. The task is defined in a form of a composition request which contains a set of available input parameters and a set of wanted output parameters. Instead of the parameter values, concepts from an ontology describing their semantics are passed to the composition engine. The composer works on a repository of services. The parameters of these services are semantically annotated in the same way as the parameters in the request. The composer then finds a set of services fulfilling the request - the composition. If the input parameters given in the request are provided, the services of this set can be executed and will finally produce the wanted output parameters. In this paper, we introduce our new, improved composition system with which we will take part in the Web Service Challenge 2008.*

13. The Web Service Challenge - A review on Semantic Web Service Composition [14]

Steffen Bleul, Thomas Weise, Kurt Geihs

Workshops der 16. ITG/ GI Fachtagung: Kommunikation in Verteilten Systemen (WowKiVS), Kassel, Germany, 2009

Abstrakt: *Every year, contesters submit contributions to the Web Service Challenge (WSC) in order to determine which service composition system is the most efficient one. In this challenge, semantic composition tasks must be solved and the results delivered by the composers are checked for correctness. The time needed for the composition process is another important competition criterion. After we had participated with great success in the 2006 and 2007 WSC, we were asked to manage the Web Service Challenge 2008. In this paper, we present the challenge task, the challenge rules, the document format used, and the results of this competition. We provide a summary over the past challenges and give first previews on the future developments planned for the Web Service Challenges to come.*

14. WSC-08: Continuing the Web Services Challenge [7]

Anjay Basal, M. Brian Blake, Srividya Kona, Steffen Bleul, Thomas Weise, Michael C. Jaeger

Proceedings of IEEE Joint Conference (CEC/EEE 2008) on E-Commerce Technology (Tenth CEC'07) and Enterprise Computing, E-Commerce and E-Services (Fifth EEE'08), 2008

15. WSC-09: A Quality of Service-Oriented Web Service Challenge [23]

Srividya Kona, Anjay Basal, M. Brian Blake, Steffen Bleul and Thomas Weise

Proceedings of IEEE Joint Conference (CEC/EEE 2009) on E-Commerce Technology (CEC'09) and Enterprise Computing, E-Commerce and E-Services (EEE'09), IEEE Computer Society, Vienna, Austria, 2009

Abstrakt: *With the growing acceptance of service-oriented computing, an emerging area of research is the investigation of technologies that will enable the discovery and composition of web services. The Web Services Challenge (WSC) is a forum where academic and industry researchers can share experiences of developing tools that automate the integration of web services. In the fifth year (i.e. WSC-09) of the Web Services Challenge, software platforms will address several new composition challenges. Requests and results will be transmitted within SOAP messages. Semantics will be represented as ontologies written in OWL, services will be represented in WSDL, and service orchestrations will be represented in WSBPEL. In addition, non-functional properties (Quality of Service) of a service will be represented using WSLA format.*

3.2 Technische Berichte

Web Service Composition Systems for the Web Service Challenge - A Detailed Review [28]

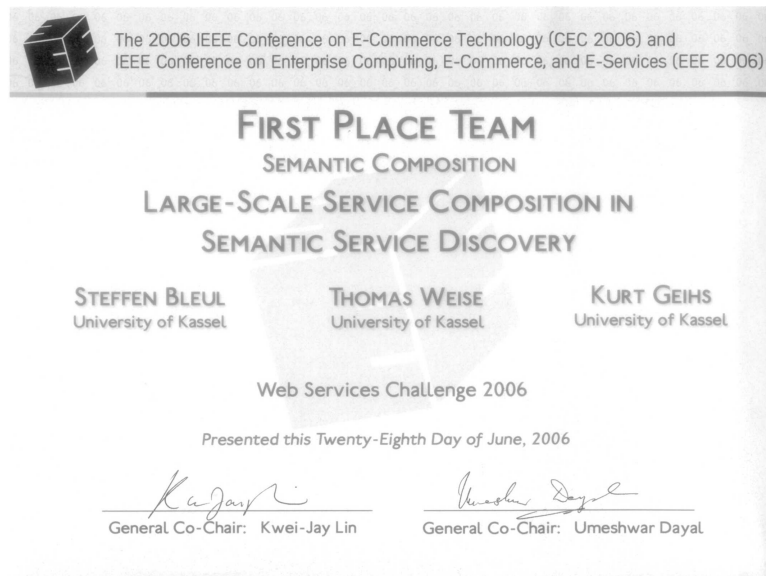
Thomas Weise, Steffen Bleul, Kurt Geihs

Kasseler Informatikschriften (KIS), Kassel, Germany, 2007

Abstrakt: *This report gives a detailed discussion on the system, algorithms, and techniques that we have applied in order to solve the Web Service Challenges (WSC) of the years 2006 and 2007. These international contests are focused on semantic web service composition. In each challenge of the contests, a repository of web services is given. The input and output parameters of the services in the repository are annotated with semantic concepts. A query to a semantic composition engine contains a set of available input concepts and a set of wanted output concepts. In order to employ an offered service for a requested role, the concepts of the input parameters of the offered operations must be more general than requested (contravariance). In contrast, the concepts of the output parameters of the offered service must be more specific than requested (covariance). The engine should respond to a query by providing a valid composition as fast as possible. We discuss three different methods for web service composition: an uninformed search in form of an IDDFS algorithm, a greedy informed search based on heuristic functions, and a multi-objective genetic algorithm.*

3.3 Preise und Auszeichnungen

Der Wettbewerb: „Web Services Challenge“



4 Fazit

ADDOaction hat seine ambitionierten Ziele im vorgegebenen Zeitrahmen erreicht und hat bewiesen, dass eine automatische Dienstvermittlung und ein automatisches Dienstmanagement durchaus möglich sind. Die erfolgreichen Ergebnisse wurden auf zahlreichen internationalen Konferenzen und Workshops veröffentlicht.

Themengebiete wie automatische Dienstvermittlung, semantische Komposition von Diensten, Management und Überwachung von Diensten und Dienstprozessen wurden ausführlich erforscht und innovative Lösungen für die offenen Probleme gefunden. Darunter fallen zahlreiche wichtige Beiträge wie die automatische Dienstvermittlung durch die semantische Erweiterung von Dienstbeschreibungen mit der neuen Mediation Contract Extension (MECE), ein erweiterter Abgleichalgorithmus, der auf MECE basiert, eine neue regelbasierte Sprache BPRules zum Service-Prozess Management, ein neuer QoS-Aggregationsalgorithmus für die Berechnung von QoS für Service-Prozesse und ein Selektionsalgorithmus für die Auswahl von Diensten.

Mit unseren semantischen Kompositionssystem haben wir erstmals 2006 beim internationalen „Web Services Challenge“ erfolgreich teilgenommen. Das Kompositionssystem haben wir in ADDOaction konsequent weiterentwickelt, so dass wir auch in den Jahren 2007 und 2008 erfolgreich waren. Im Jahre 2008 und 2009 waren wir sogar selbst mit der Organisation des Wettbewerbs beauftragt.

Das ADDOaction Framework umfasst eine Vielzahl an Softwarekomponenten, welche prototypisch realisiert wurden, um die Machbarkeit der Konzepte zu demonstrieren. Realitätsnahe Szenarien wurden entwickelt und getestet, um die wichtigsten Forschungsergebnisse zu veranschaulichen.

Die aktuellen Forschungsthemen von ADDO und ADDOaction haben Interesse auch bei den Studenten geweckt. So fanden Seminare zum Thema „Service Oriented Architecture“ (SOA) statt. Auch konnten wir mehrere Studenten für Dienstorientierte Architekturen begeistern, so dass einige Projektarbeiten und Bachelorarbeiten unter unserer Betreuung geschrieben wurden.

Die Forschungsergebnisse von ADDOaction können sowohl in Industrie- als auch in Forschungsprojekten (zum Beispiel in den Projekten MUSIC und VENUS) eingesetzt und weiter ausgebaut werden. ADDOaction dient hier als Basis für weiterführende Arbeiten im interdisziplinären Umfeld von VENUS, wo Dienste in einer ubiquitären Umgebung zusammenspielen und auch Aspekte aus Sicht von Recht-, Ergonomie- oder Wirtschaftswissenschaften untersucht werden.

Literaturverzeichnis

- [1] Automated Service Brokering in Service-oriented Architectures (ADDO) in Aktion Projekt.
URL: <http://www.vs.uni-kassel.de/research/addo/>, Letzter Zugriff 8.03.2010.
- [2] EU IST FP6 project MUSIC (Self-adapting Applications for Mobile Users in Ubiquitous Computing Environments).
URL: <http://www.ist-music.eu>, Letzter Zugriff 8.03.2010.
- [3] VENUS Projekt (Gestaltung technisch-sozialer Vernetzung in situativen ubiquitären Systemen).
URL: <http://cms.uni-kassel.de/unicms/index.php?id=venus>, Letzter Zugriff 8.03.2010.
- [4] Web Service Challenge 2007.
URL: <http://ws-challenge.georgetown.edu/wsc07/>, Letzter Zugriff 8.03.2010.
- [5] Web Service Challenge 2008.
URL: <http://cec2008.cs.georgetown.edu/wsc08/>, Letzter Zugriff 8.03.2010.
- [6] Tony Andrews, Francisco Curbera, Hitesh Dholakia, Yaron Goland, Johannes Klein, Frank Leymann, Kevin Liu, Dieter Roller, Doug Smith, Satish Thatte, Ivana Trickovic, and Sanjiva Weerawarana. *BPEL4WS, Business Process Execution Language for Web Services Version 1.1*. IBM, 2003.
- [7] Anjay Basal, M. Brian Blake, Srividya Kona, Steffen Bleul, Thomas Weise, and Michael C. Jaeger. Wsc-08: Continuing the web services challenge. In *Proceedings of IEEE Joint Conference (CEC/EEE 2008) on E-Commerce Technology (Tenth CEC'07) and Enterprise Computing, E-Commerce and E-Services (Fifth EEE'08)*, pages 351–354. IEEE Computer Society, 2008.
- [8] Steffen Bleul. Quality-aware self-management for service processes in service-oriented architectures. In Tudor Dumitras, Andreas Hanemann, Benedikt Kratz, and Jyotishman Pathak, editors, *IBM PhD Student Symposium at ICSSOC 2007*, Vienna, Austria, September 2007. IBM Research Division.
- [9] Steffen Bleul, Diana Comes, Marc Kirchhoff, and Michael Zapf. Self-integration of web services in bpel processes. In *Proceedings of the Workshop Selbstorganisierende, Adaptive, Kontextsensitive verteilte Systeme (SAKS)*, 2008.
- [10] Steffen Bleul and Kurt Geihs. Automatische dienstvermittlung in dienstorientierten architekturen. Technical Report 34-2007070, Kasseler Informatikschriften (KIS), Kassel, Germany, 2007.
- [11] Steffen Bleul, Kurt Geihs, Diana Comes, and Marc Kirchhoff. Automated Management of Dynamic Web Service Integration. In *Proceedings of the 15th Annual Workshop of HP Software University Association (HP-SUA)*. Infocomics-Consulting, Stuttgart, Germany, 2008.
- [12] Steffen Bleul, Kurt Geihs, Diana Comes, and Marc Kirchhoff. Automated integration of web services in bpel4ws processes. In *16. Fachtagung Kommunikation in Verteilten Systemen, KiVS 2009, Kassel*, Kassel, Germany, march 2009. University of Kassel, VDE.
- [13] Steffen Bleul, Thomas Weise, and Kurt Geihs. Making a Fast Semantic Service Composition System Faster. In *Proceedings of the 2007 IEEE Joint Conference on E-Commerce Technology (CEC'07) and Enterprise Computing, E-Commerce and E-Services (EEE'07)*, Tokyo, Japan, July 2007.

- [14] Steffen Bleul, Thomas Weise, and Kurt Geihs. The web service challenge - a review on semantic web service composition. In Michal Wagner, Dieter Hogrefe, Kurt Geihs, and Klaus David, editors, *Service-Oriented Computing (SOC'2009)*. The European Association of Software Science and Technology, 2009.
- [15] Steffen Bleul, Michael Zapf, and Kurt Geihs. Automatic Service Process Administration by Semantic Service Discovery. In *Proceedings of the 7th International Conference on New Technologies of Distributed Systems*, June 2007.
- [16] Steffen Bleul, Michael Zapf, and Kurt Geihs. Flexible Automatic Service Brokering for SOAs. In *Proceedings on 10 th IFIP / IEEE Symposium on Integrated Management (IM 2007)*, Munich, Germany, May 2007.
- [17] D. Booth and C.K. Liu. *WSDL, Web Services Description Language*. World Wide Web Consortium, August 2007.
- [18] J. Clark. *XSL Transformations (XSLT) Version 1.0, W3C Recommendation*. World Wide Web Consortium, November 1999.
- [19] Diana Comes, Steffen Bleul, Thomas Weise, and Kurt Geihs. A Flexible Approach for Business Processes Monitoring. In *Proceedings of the 9th IFIP international conference on Distributed Applications and Interoperable Systems (DAIS 2009)*. Springer, 2009.
- [20] Diana Comes, Steffen Bleul, and Michael Zapf. Management of business processes with the bprules language in service oriented computing. In *Workshops der Wissenschaftlichen Konferenz Kommunikation in Verteilten Systemen (WowKiVS) 2009, Kassel, Kassel, Germany, 2009*. University of Kassel, VDE.
- [21] A. Alves A. Arkin S. Askary C. Barreto B. Bloch F. Curbera M. Ford Y. Goland A. Guizar N. Kartha C.K. Liu R. Khalaf D. König M. Marin V. Mehta S. Thatte D. van der Rijn P. Yendluri D. Jordan, J. Evdemon and A. Yiu. *Web Services Business Process Execution Language Version 2.0*. OASIS, 2007.
- [22] Kurt Geihs, Roland Reichle, Michael Wagner, and Mohammad Ullah Khan. Modeling of context-aware self-adaptive applications in ubiquitous and service-oriented environments. In *Software Engineering for Self-Adaptive Systems*, pages 146–163, Berlin, Heidelberg, 2009. Springer-Verlag.
- [23] Srividya Kona, Ajay Bansal, M. Brian Blake, Steffen Bleul, and Thomas Weise. Wsc-2009: A quality of service-oriented web services challenge. In *Proceedings of the 11th IEEE Conference on Commerce and Enterprise Computing (CEC'09)*. IEEE Computer Society, 2009.
- [24] B. Miller. *The autonomic computing edge: The role of knowledge in autonomic systems*, 2009.
- [25] E. Miller and J. Hendler. *Web Ontology Language (OWL)*. World Wide Web Consortium (W3C), 2004.
- [26] OASIS. *The Universal Description, Discovery and Integration (UDDI) protocol. Version 3*. Organization for the Advancement of Structured Information Standards, 2003.
- [27] Thomas Weise, Steffen Bleul, Diana Comes, and Kurt Geihs. Different approaches to semantic web service composition. In *Proceedings of The Third International Conference on Internet and Web Applications and Services, ICIW 2008*. IEEE Computer Society Press, 2008.

- [28] Thomas Weise, Steffen Bleul, and Kurt Geihs. Web service composition systems for the web service challenge - a detailed review. Technical Report 34-2007111919638, November 2007. Permanent Identifier: urn:nbn:de:hebis:34-2007111919638.
- [29] Thomas Weise, Steffen Bleul, Marc Kirchhoff, and Kurt Geihs. Semantic web service composition for service-oriented architectures. In *Proceedings of IEEE Joint Conference (CEC/EEE 2008) on E-Commerce Technology (Tenth CEC'07) and Enterprise Computing, E-Commerce and E-Services (Fifth 08)*, pages 355–358. IEEE Computer Society, 2008.