

Zwischenbericht

DFG-Projekt



Automatische Dienstvermittlung in
dienstorientierten Architekturen

Kurt Geihs und Steffen Bleul

Universität Kassel



VERTEILTE SYSTEME

1 Zusammenfassung

1.1 Darstellung der wesentlichen Ergebnisse und der Fortschritte.

Die Technologie dienstorientierter Architekturen (*Service-oriented Architectures*, kurz *SOA*) weckt große Visionen auf Seiten der Industrie wie auch der Forschung. Sie hat sich als derzeit ideale Lösung für Umgebungen, in denen sich die Anforderungen an die IT-Bedürfnisse rapide ändern, erwiesen. Heutige IT-Systeme müssen Managementaufgaben wie Softwareinstallation, -anpassung oder -austausch erlauben, ohne dabei den laufenden Betrieb wesentlich zu stören. Die dafür nötige Flexibilität bieten dienstorientierte Architekturen, in denen Softwarekomponenten in Form von *Diensten* zur Verfügung stehen.

Ein Dienst bietet über seine Schnittstelle lokalen wie entfernten Applikationen einen Zugang zu seiner Funktionalität. Wir betrachten im Folgenden nur solche dienstorientierte Architekturen, in denen Dienste zur Laufzeit dynamisch entdeckt, gebunden, komponiert, verhandelt und adaptiert werden können. Eine Applikation kann mit unterschiedlichen Diensten arbeiten, wenn beispielsweise Dienste ausfallen oder ein neuer Dienst die Anforderungen der Applikation besser erfüllt. Eine unserer Grundvoraussetzungen lautet somit, dass sowohl das Dienstangebot als auch die Nachfrageseite variabel sind.

Dienstorientierte Architekturen haben besonderes Gewicht in der Implementierung von Geschäftsprozessen. Im Rahmen des Paradigmas *Enterprise Integration Architecture* werden einzelne Arbeitsschritte als Dienste implementiert und ein Geschäftsprozess als Workflow von Diensten ausgeführt. Eine solche Dienstkomposition wird auch Orchestration genannt. Insbesondere für die so genannte B2B-Integration (Business-to-Business) sind Dienste das probate Mittel, um die Kommunikation über die Unternehmensgrenzen hinaus zu unterstützen. Dienste werden hier in der Regel als *Web Services* realisiert, welche vermöge *BPEL4WS* orchestriert werden. Der XML-basierte Nachrichtenverkehr und das http-Protokoll sorgen für eine Verträglichkeit zwischen heterogenen Systemen und eine Transparenz des Nachrichtenverkehrs.

Anbieter dieser Dienste versprechen sich einen hohen Nutzen durch ihre öffentlichen Dienste. Zum einen hofft man auf eine vermehrte Einbindung ihrer Dienste in Softwareprozesse. Zum anderen setzt man auf das Entwickeln neuer Software auf Basis ihrer Dienste. In der Zukunft werden hunderte solcher Dienste verfügbar sein und es wird schwer für den Entwickler passende Dienstangebote zu finden.

Das Projekt ADDO hat in diesem Umfeld wichtige Ergebnisse erzielt. Im Laufe des Projektes wurde erreicht, dass der Einsatz semantischer Spezifikationen es ermöglicht, Dienste sowohl im Hinblick auf ihre funktionalen als auch ihre nicht-funktionalen Eigenschaften, insbesondere die Dienstgüte, automatisch zu sichten und an Dienstaggregate zu binden [15]. Dazu wurden Ontologie-Schemata [10, 16], Abgleichalgorithmen [16, 9] und Werkzeuge entwickelt und als Framework implementiert [16]. Der in diesem Rahmen entwickelte Abgleichalgorithmus für Dienstgüte beherrscht die automatische Aushandlung von Verträgen für die Dienstnutzung, um etwa kostenpflichtige Dienste zur Dienstnutzung einzubinden.

ADDO liefert einen Ansatz, Schablonen für Dienstaggregate in BPEL4WS zu erstellen, die zur Laufzeit automatisch verwaltet werden. Das Vorgehen konnte seine Effektivität beim internationalen Wettbewerb *Web Service Challenge 2006* in San Francisco unter Beweis stellen: Der für ADDO entwickelte Algorithmus zur semantischen Dienstkomposition erreichte den ersten Platz. Der Algorithmus erlaubt es, unter einer sehr großen Menge angebotener Dienste eine geeignete Auswahl zu treffen, diese Dienste zu Dienstaggregaten zusammenzufassen und damit die Funktionalität eines vorgegebenen gesuchten Dienstes zu leisten. Weitere Ergebnisse des Projektes ADDO wurden auf internationalen Workshops und Konferenzen veröffentlicht. [12, 11]

1.2 Ausblick auf künftige Arbeiten

Künftige Arbeiten werden im Nachfolgeprojekt ADDOaction durchgeführt. Ziel der bereits genehmigten zweiten Phase von ADDO ist es, die gefundenen Ansätze weiter auszubauen, um das Management von Dienstaggagaten in einer dienstorientierten Umgebung zu automatisieren und die Machbarkeit auf Basis der standardisierten Komponentenplattform OSGi zu demonstrieren. OSGi findet heute Anwendung in vielen unterschiedlichen Szenarien und auf einer breiten Palette von Geräten. Beispiele für OSGi-Anwendungen sind die Softwareentwicklungsumgebung Eclipse [2], TV Set-top Box [19], Gebäudeautomatisierung [23] und Kraftfahrzeugsteuerung [7]. Es wundert daher nicht, dass OSGi immer größere Beachtung bei der Erforschung und Entwicklung softwareintensiver Produkte erfährt. OSGi-Implementierungen sind als Open Source verfügbar [3, 4, 1]. Kernpunkte der Forschungsarbeiten von ADDOaction sind:

- Definition einer verteilten **Dienstinfrastruktur** auf Basis der ADDO-Ergebnisse, die das dienstgüteorientierte Selbst-Management von Diensten und Dienstaggagaten ermöglicht. Das schließt Aufgaben wie Selbst-Heilung, Selbst-Adaption und Selbst-Optimierung mit ein.
- **Prototypische Realisierung** der neuen Dienstinfrastruktur auf dem standardisierten verteilten Komponenten-Framework OSGi.
- Automatische **Code-Generierung** der für das Selbst-Management benötigten Sensoren, Monitore und Aktoren anhand semantischer Modelle, die – wie bei ADDO – ein wichtiger Bestandteil des Frameworks sind. Der erzeugte Code wird zu OSGi-spezifischen Bundles kompiliert und zur Laufzeit eingebracht, wenn es eine sich ändernde Dienstkonfiguration erfordert.
- Prototypische Implementierung einer realistischen **Anwendung** und Evaluation der neuen Dienstinfrastruktur und ihrer Selbst-Eigenschaften.

Die Arbeitsgruppe des Antragstellers hat mit dem sehr erfolgreichen Vorgängerprojekt ADDO eine hervorragende Basis, um die ambitionierten Forschungsziele von ADDOaction zu erreichen. Es ist dabei eine durchaus realistische Annahme, dass sich daraus mittelfristig auch Potenziale für die Standardisierung (z.B. bei OSGi) und für die Vermarktung innovativer Produkte ergeben.

2 Arbeiten und Ergebnisse

2.1 Ausgangslage

Anwendungen in Unternehmen werden heute im Gegensatz zu den isolierten Einzellösungen der Vergangenheit meist als Aggregate weitgehend eigenständiger Softwarekomponenten realisiert. Dienstorientierte Architekturen sind die Basis für die Erweiterung dieser Aggregate um extern erbrachte Dienstkomponenten, die bevorzugt über das Internet als Web Services eingebunden werden. Damit wird es beispielweise möglich, betriebswirtschaftliche Anwendungen zu Ablaufketten zu verbinden, externe Informationsdienste in die eigene Planungssoftware einzu beziehen oder ein netzweites Single-Sign-On zu nutzen. Solche dienstorientierten Architekturen finden heute große Aufmerksamkeit, weil sich die Anwender dadurch Flexibilität, Interoperabilität, Anpassungsfähigkeit, Wiederverwendbarkeit, Herstellerunabhängigkeit und letztendlich Kostenersparnisse für die Entwicklung und den Betrieb der Geschäftsanwendungen erhoffen.

Damit dienstorientierte Architekturen Realität werden können, sind standardisierte Techniken für die Beschreibung und die Kopplung der Komponenten erforderlich. Hier findet heute die Technik der Web Services große Beachtung. Web Services stellen Softwarekomponenten dar, die über das Internet angeboten, gefunden und genutzt werden können. Es verbindet sich damit die Hoffnung, externe Komponenten auf der Basis offener Internet-Standards, wie XML und HTTP, zu integrieren. Web Services bieten nicht mehr - bisher eher weniger - Funktionalität als Verteilungsplattformen wie CORBA, J2EE und .NET; ihre Attraktivität rührt vor allem daher, dass sie von allen wichtigen Herstellern als gemeinsamer Nenner für die Anwendungsintegration akzeptiert und gefördert werden. Die Web Services-Technik ist somit auf dem Weg, das Internet zur Middleware für dienstorientierte Architekturen werden zu lassen. Bis dahin ist es aber noch ein weiter Weg, da die existierenden Standards nur eine sehr eingeschränkte Verteilungsinfrastruktur darstellen.

Eines der bisher ungelösten Probleme ist das automatische Konfigurieren und Binden der Komponenten. Ganz ähnlich zu der Selbstkonfiguration, die beispielsweise bei der Benutzung eines neuen Gerätes an der USB-Schnittstelle eines PC stattfindet, sollen Anwendungen in der Lage sein, Dienstangebote dynamisch einzubinden und zu nutzen. Neue Entwicklungen wie Semantic Web, Dienstgütegarantien und Web Services haben die technischen Randbedingungen grundsätzlich verändert. Sie bieten ein enormes Verbesserungspotential, werfen aber auch eine Reihe interessanter offener Forschungsfragen auf. Ziel muss es sein, so viel syntaktische und semantische Informationen zur Beschreibung der Dienstkomponenten bereit zu stellen, damit passende Dienstangebote zur Laufzeit gefunden und dynamisch gebunden werden können, d.h. damit sich die Anwendung selbst konfigurieren kann. Frühere Ansätze zur Dienstvermittlung (z.B. der CORBA Trader) basierten auf syntaktischen Schnittstellenbeschreibungen und der Verwendung einfacher Dienstattribute. Dieser Ansatz ist völlig unzulänglich, um die Interoperabilität und Selbstkonfiguration von Komponenten zu erreichen.

Für komplexe Anwendungsszenarien werden die Kunden nicht nur einzelne Dienste sondern auch die Verkettung unabhängiger Dienste „als Paket“ nachfragen. Auch werden sich einzelne Dienstanbieter mit anderen zu Partnerschaften zusammenschließen, um gemeinsam einen höherwertigen Verbunddienst anzubieten. Zum Beispiel wäre es in einem Unternehmen sehr gut vorstellbar, dass die Einzel-Dienste „Bestellannahme“, „Lagerhaltung“, „kostengünstigster Anbieter“ und „Kreditkartenzahlung“ zu einem Verbunddienst „Bestellungsabwicklung“ kombiniert werden. Diese Verkettung sollte aus Sicht des Klienten einfach zu bewerkstelligen sein und bei der Benutzung weitgehend transparent bleiben. Folglich soll auch die Vermittlung von Dienstaggregaten bei der Dienstvermittlung adäquat unterstützt werden.

Ziel des Forschungsvorhabens ADDO war es, auf den eigenen Vorarbeiten im Bereich des Service Trading aufzubauen und das Problem der Vermittlung von Dienstkomponenten unter Beachtung der neuen technischen Randbedingungen zu lösen. Kernpunkte der Forschungsarbeiten waren:

- Definition einer **Ontologie** zur Spezifikation von Diensteseigenschaften, einschließlich der Dienstgüteeanforderungen auf Basis der semantischen Dienstbeschreibungssprache OWL-S.
- Entwicklung eines flexiblen, benutzerspezifisch konfigurierbaren **Abgleichalgorithmus** für den Vergleich von Dienstbeschreibungen.
- Automatisierte Vermittlung von **Dienstaggregaten**, wobei die Dienstkomposition transparent für den nachfragenden Klienten erfolgt.
- **Prototypische Implementierung** auf der Basis von Web Services und die Erprobung in einem realistischen Anwendungsszenario.

Die Automatisierung des Bindevorgangs ist ein wichtiger Baustein für die Selbstorganisation und Selbstkonfiguration in verteilten Systemen.

2.2 Beschreibung der durchgeführten Arbeiten

Das Forschungsprojekt ADDO hat erfolgreich ein Framework für die automatische Dienstvermittlung realisiert, welches das automatische Sichten und Binden von Diensten ermöglicht. Die automatische Dienstvermittlung basiert auf Abgleichalgorithmen, die auf semantischen Beschreibungen arbeiten, und einer Infrastruktur zur automatischen Bindung. Innerhalb der Projektzeit wurde, wie geplant, das bestehende Typsystem OWL-S angepasst und damit die Dienstsichtung und Selbst-Konfiguration für einen einzelnen Dienst konzeptioniert und implementiert. Darüber hinaus konnte im Projekt die bisherige Forschungslücke, die Berücksichtigung von Dienstgüte, geschlossen werden. Die Ergebnisse und Erfahrungen haben geholfen, in der zweiten Phase Dienstaggregate zu berücksichtigen und den Schritt von Diensten zu Dienstaggregaten erfolgreich durchzuführen.

Schon in der Analysephase hat sich gezeigt, dass das semantische Typsystem OWL-S zwar ein geeignetes Schema für Dienste, insbesondere Web Services, darstellt, aber für die automatische Dienstvermittlung Erweiterungen notwendig sind. Daher wurden zwei Erweiterungen geschaffen. Zuerst wurden die bestehenden Schemata um die semantische Spezifikation von Dienstgüte erweitert und darauf aufbauend sogleich Schemata für Dienstgüteeverträge definiert. Dabei orientierten wir uns an den syntaktischen Sprachen WSLA von IBM [24], WSML von HP [27] und der Sprache WS-A [25], die aus den Forschungen der Grid Community entstand. In Anlehnung an die syntaktischen Vorbilder ist eine Ontologie entstanden, die zusätzlich zur syntaktischen Information semantische Deduktion erlaubt. Die entstandene Ontologie wurde in [10, 11] veröffentlicht.

Danach musste die Ausdrucksfähigkeit von OWL-S bezgl. der Spezifikation von funktionalen Aspekten eines Dienstes erweitert werden. Der Stand der Forschung beschränkte sich bis zu dem Zeitpunkt auf halb-automatische Dienstvermittlung. Das beschränkte die Dienstsichtung auf Abgleichalgorithmen, die auf der Eingabe von Spezialisten arbeiteten. Der Algorithmus traf automatisch eine Vorauswahl, die vom Administrator dann überprüft und manuell übernommen werden musste. Eine automatische Dienstsichtung kann nur dann erfolgreich zur Selbst-Konfiguration führen, wenn die Dienstanfrage alle notwendigen Information enthält, die dann auch entsprechend vom Abgleichalgorithmus berücksichtigt werden. In ADDO werden sowohl die Dienstanfragen als auch die Dienstangebote mit semantischen Schnittstellenbeschreibungen spezifiziert.

Als Resultat entstand in Anlehnung an OWL-S ein Ontologieschema, das strikt zwischen einem Dienstangebot und einer Dienstanfrage unterscheidet. Jede Dienstanfrage und jedes Dienstangebot wird einmalig von einem Spezialisten spezifiziert und von einem Algorithmus zur Dienstsichtung gegeneinander abgeglichen. Da hinsichtlich der Dienstgüte eine neue Ontologie entstand, musste auch der Abgleichalgorithmus zur Sichtung von Dienstgüteeanforderungen erweitert werden. Darüber hinaus führte diese Unterscheidung von Dienstanfrage

und Dienstangebote zu einem flexiblen und benutzerspezifisch konfigurierbaren Abgleichalgorithmus, weil die Schemata auch Informationen enthalten, die entsprechend beim Ranking der Ergebnisse berücksichtigt werden. Zu diesem Zeitpunkt hatten wir bereits die notwendigen Grundlagen für eine automatische Dienstsichtung in ADDO geschaffen und das Ergebnis wurde im Anschluss in [16, 15] veröffentlicht.

Ein weiteres Ziel des Projektes war es, die Forschungsarbeit auf die Sichtung von Dienstaggregaten auszuweiten. Dabei werden Dienstaggregate ermittelt, die als einheitliche Kombination und bei entsprechender Ausführung in der richtigen Reihenfolge die gleiche Funktionalität wie ein einzelner Dienst bieten. Gegenüber der Sichtung eines einzelnen Dienstes ergibt sich das Problem, dass zusätzlich zur Auswertung funktionaler Aspekte alle möglichen Kombinationen von Diensten berücksichtigt werden müssen. Für die Dienstsichtung müssen dann auch der Algorithmus und die Implementierung sehr effizient arbeiten. Zu diesem Zweck wurden ein Kompositionsalgorithmus und Kompositionssystem entwickelt. Das entstandene Kompositionssystem wurde implementiert und veröffentlicht. Es nahm an der Web Service Challenge 2006 teil und errang dort den ersten Platz [12].

Anschließend wurden die Arbeiten in ADDO auf ein aussagekräftiges Szenario für den Einsatz in einem Unternehmen zugeschnitten. In einem Unternehmen existieren klare Zielvorstellungen für den Einsatz einer automatisierten Dienstvermittlung. Arbeitsprozesse werden durch eine Abfolge von Einzeldiensten implementiert, wobei für die Definition solcher Abfolgen Ausführungssprachen wie BPEL4WS existieren. Wenn ein Dienst ausfällt, besteht die Gefahr, dass der gesamte Prozess seine Verfügbarkeit einbüßt. Für jeden Dienstprozess sollen zu jedem Zeitpunkt die notwendigen Dienste zur Verfügung stehen. Zu diesem Zweck wurde im Rahmen des Projektes eine Ontologie für Dienstprozesse entwickelt, um semantische Dienstsichtung für Dienstprozesse zu ermöglichen. Die Ontologie basiert auf dem Grundgedanken, dass Geschäftsprozesse, obwohl die Dienste, die daran teilnehmen, einer Dynamik unterliegen, weitgehend statisch sind. Die Dienstprozesse können sich z.B. in den Bezeichnern und Datentypen unterscheiden, die in ihren Schnittstellenbeschreibungen verwendet werden. Das liegt daran, dass in einem Unternehmen unterschiedliche Dienstanbieter und verschiedene Versionen von Diensten vorhanden sein können. Daher ist es notwendig, nicht nur einen funktional gleichwertigen Dienst zu finden, sondern sogar eine passende Prozessbeschreibung. Die durchgeführte Arbeit wurde durch einen Prototypen verifiziert und in [16, 15] veröffentlicht. Abschließend haben wir in [14] Selbst-Eigenschaften beschrieben, die durch unsere Forschungsergebnisse realisiert werden können.

Zusammenfassend kann man sagen, dass die Arbeiten, wie zu Projektstart festgelegt, innerhalb des gegebenen Zeitrahmens ausgeführt und die oben genannten Herausforderungen und Probleme bewältigt wurden. Da aber nur eine von ursprünglich zwei beantragten Wissenschaftlerstellen für ADDO bewilligt wurden, konnten manche Probleme nicht in der gewünschten Tiefe behandelt werden. So erscheint bei den Abgleichalgorithmen das Potential semantischer Zusatzinformation noch nicht vollkommen ausgeschöpft. Auch konnte die Implementierung nicht im ursprünglich geplanten Umfang durchgeführt werden. Beides schmälert aber den Wert der erzielten Ergebnisse nur unwesentlich. Die Ergebnisse des Projektes sind viel versprechend und ergeben neue Forschungsfragen und Möglichkeiten für die Selbst-Konfiguration und das Management von SOAs. Die Forschungsarbeiten werden deswegen im Rahmen des Folgeprojektes ADDOaction weitergeführt.

2.3 Darstellung der erzielten Ergebnisse

ADDO hat erfolgreich die Grundlage für die automatische Dienstvermittlung gelegt und damit eine Säule einer selbst-organisierenden dienstorientierten Architektur realisiert. Ein Dienst wird in ADDO - wie allgemein üblich - zunächst bezüglich seiner Funktionalität gesichtet. Es werden so Dienste gefunden, die sich generell für eine vorgegebene Aufgabe eignen. Darüber hinaus betrachtet die ADDO-Lösung die Dienste bezüglich ihrer Dienstgüte. So wird es erst möglich, dass nur die Dienste ausgewählt werden, die auch die Leistungsanforderungen erfüllen. Dazu wurde die bereits verfügbare OWL-S Ontology [18] für Dienstschnittstellen um die im Projekt entstandene SL-Ontologie erweitert [10]. Diese Ontologie wird dazu verwendet, ein Dienstangebot mit einer Spezifikation ihrer Dienstgüte zu versehen.

Für die Sichtung von Diensten wurden in ADDO neue Abgleichalgorithmen entwickelt, die auch Service Level Agreements berücksichtigen können. Damit können dann sogar vertragspflichtige bzw. kostenpflichtige Dienste automatisch gesichtet, vermittelt und gebunden werden. Die Analyse in ADDO hat gezeigt, dass existierende semantische Beschreibungssprachen und Projekte, wie z.B. METEOR-S [26], AMIGO [22] und ODE-SWS [21], diesen Vorgang nur unzureichend unterstützen. Gerade auch wegen dieser Neuerung wurde unser Konferenzbeitrag über die SL-Ontologie dazu ausgewählt, in erweiterter Fassung in einer Sonderausgabe des CSSE Journals abgedruckt zu werden [11]. Neben unserer SL-Ontologie wurden zusätzliche Elemente wie multiple Operationen und Effekte von Operationen in OWL-S eingefügt. Von besonderer Bedeutung sind die neuen Service Request und Result Ontologien. Damit werden Anfrage, Sichtung und Ergebnis in einer Wissensbasis festgehalten und für jede Art der Wiederverwendung zur Verfügung gestellt. Eine detaillierte Betrachtung dieser Vorgehensweise findet sich in [16].

Ausserdem ist im Projekt erstmals eine Lösung und ein Prototyp für Dienstprozesse entstanden [15], um semantische Dienstsichtung für ganze Dienstprozesse mit BPEL4WS zu ermöglichen. Bisherige Lösungen, z.B. RobustBPEL2 [20], BPEL4J [17] und [8], berücksichtigen nicht die Möglichkeiten der semantischen Dienstsichtung. Die Ontologie beschreibt die notwendigen Rollen, für die Dienste notwendig sind, während die Ausführung durch die Prozessbeschreibung BPEL4WS definiert wird. Dadurch ist es möglich, dass der Prototyp die Verfügbarkeit aller beteiligten Rollen sicherstellt.

ADDO hat aber nicht nur für fest spezifizierte Kompositionen eine Lösung geschaffen sondern auch für die automatische Komposition von Diensten zu Dienstaggagaten. Der in ADDO entwickelte Algorithmus zur semantischen Komposition von Diensten ist der schnellste seiner Art. Damit errang die Gruppe des Antragstellers beim internationalen Wettbewerb „Web Service Challenge 2006 (WSC06)“ den ersten Platz unter acht Teams [5]. Ebenso wurde das System anhand von genetischen Algorithmen erweitert [13]. Auch dieses Verfahren wurde implementiert, veröffentlicht und nimmt an der „Web Service Challenge 2007“ teil [6].

Insgesamt zeichnet sich ADDO nicht nur durch seine konzeptionellen Ergebnisse aus, sondern auch durch die prototypische Implementierung. Das belegen sowohl der oben genannte, erfolgreiche Kompositionsalgorithmus als auch das realisierte System in [12]. Erstmals wurde in dieser Form gezeigt, wie ein System zur automatischen Dienstvermittlung in einer dienstorientierten Architektur eingesetzt werden kann. Die Implementierung unseres Dienstvermittlers ist mit aktuellen Techniken der Softwareentwicklung entstanden und kann deshalb ohne Probleme in bestehende Systeme integriert werden.

In [14] werden die Selbst-Eigenschaften diskutiert, die mit Hilfe einer automatischen Dienstvermittlung, wie durch ADDO zur Verfügung gestellt, erreicht werden können. Selbst-Konfiguration/Adaption, Selbst-Heilung und Selbst-Optimierung werden nun im Folgeprojekt ADDOaction realisiert.

3 Ausblick

3.1 Ausblick auf künftige Arbeiten

Im Laufe des Projektes haben sich einige Forschungsziele ergeben, die im Folgeprojekt ADDOaction behandelt werden sollen. Dienstorientierte Architekturen sind derzeit die ideale Lösung für Umgebungen, in denen sich die Anforderungen an die IT-Bedürfnisse rapide ändern. Heutige IT-Systeme müssen Managementaufgaben wie Softwareinstallation, -anpassung oder -austausch erlauben, ohne dabei den laufenden Betrieb wesentlich zu stören. Das Vorgängerprojekt ADDO hat mit der Realisierung der automatischen Dienstvermittlung hier wichtige Grundlagen gelegt.

Mit dienstorientierten Architekturen lässt sich eine Konfigurationsflexibilität und Anwendungsadaptivität prinzipiell erreichen. Automatische Lösungen stehen aber zurzeit noch nicht zur Verfügung. Die angestrebte Dynamik erfordert von solchen Systemen ein hohes Maß an Automatismen, d.h. Selbst-Management. Sobald es sich um kommerzielle Dienste handelt, stellt sich sogleich die Frage nach den Anforderungen an das System in Bezug auf die Vertragsverhandlung und den Vertragsabschluss, die Form dieser Verträge und die Kostenabwicklung sowie die Überwachung der Einhaltung von Verträgen. Die Qualitätsmerkmale eines Dienstaggregats und seiner Dienste müssen fortlaufend überwacht werden, sei es zur Wahrung kommerzieller Vertragsbedingungen oder in einem dienstgütekritischen Bereich, zum Beispiel in einem Krankenhaus. Diese Überwachung erfordert eine geeignete Systemarchitektur, welche entsprechende Messpunkte und Qualitätskriterien für die beteiligten Dienste kennt und auf dieser Basis die Einhaltung der Vertragsbedingungen und das einwandfreie Arbeiten des Dienstaggregates sicherstellt.

Außerdem ist zu untersuchen, ob es möglich ist, das Management einer SOA auszulagern, ohne die Interessen des Unternehmens zu verletzen oder internes Wissen der Firma offen zu legen. Dienste und Dienstprozesse sind die Bausteine, aus denen in der Zukunft die Geschäftsprozesse eines Unternehmens bestehen werden. Unternehmen können ihre IT-Kosten für Entwicklung und Verwaltung durch geschicktes Einkaufen von Diensten minimieren. Bisher wurde nur die Frage behandelt, ob man Dienste kaufen und integrieren kann. Aber nicht nur das Know-How über eine spezielle Geschäftslogik, wie bei einem Kreditkartendienst, lässt sich verkaufen, sondern auch das Know-How für das Management von Diensten und speziell Dienstprozesse.

3.2 Interdisziplinäre Weiterentwicklung

Die erarbeiteten Ergebnisse lassen sich in allen Bereichen einsetzen, in denen bereits Dienste oder Dienstleistungen in Verbindung mit Dienstgüte verwendet werden, z.B. in den Forschungsgebieten Grid Computing und Ubiquitous Computing.

Grid Computing untersucht den Nutzen und Verwendung von verteilten Ressourcen. Dabei handelt sich sowohl um Hardware-Ressourcen, wie Großrechner, Server und Speichermedien, als auch um Softwaresysteme, wie Datenbanken und aufwändige numerische Berechnungspakete. Insbesondere der kommerzielle Nutzen, der durch die stete Nachfrage nach Ressourcen entsteht, macht diesen Bereich so interessant. ADDO liefert für Grid Computing zwei anwendbare Lösungen. Zum einen sind die angebotenen Ressourcen nur in Verbindung mit Dienstgüte zu verkaufen, zum anderen sind die Ressourcen durch Dienstschnittstellen zu nutzen. Im Grid-Computing entstanden die Sprache und das Modell für Dienstgüte namens Web Service Agreement (WS-A) [25]. Da in ADDO die Dienstgüte ein Forschungsschwerpunkt war und die entstandene Ontologie auf Aspekten von WS-A aufbaut, lässt sich sowohl die Ontologie als auch der entstandene Abgleichalgorithmus wieder verwenden.

Darüber hinaus lässt sich das Management von Grid-Ressourcen automatisieren, sofern für jede Ressource Management-Schnittstellen existieren, die selbst wieder als Dienste verfügbar sind. Durch automatische Dienstvermittlung, wie in ADDO, können die Management-

Endpunkte gesichtet und genutzt werden. Eine Applikation des Dienstnutzers, die für das Management von Grid-Ressourcen zuständig ist, kann so Ressourcen sichten und die angebotene Managementfunktionalität ansprechen.

Ubiquitous Computing geht von der Annahme aus, dass die Informationsverarbeitung mit zunehmendem Maße für den Benutzer allgegenwärtig wird. Mobile Geräte wie Mobiltelefone, PDA und Laptop gehören mittlerweile zum ständigen Begleiter vieler Menschen. Zusätzlich werden in zunehmendem Maße verteilte Systeme eingesetzt, um z.B. das eigene Heim, den Arbeitsplatz oder öffentliche Plätze mit entsprechenden Diensten zu automatisieren. Ein erster Ansatz wurde in Zusammenarbeit mit dem C-Lab der Universität Paderborn untersucht und in [28] veröffentlicht. Eine der wichtigsten Fragestellungen ist, entsprechende Dienste miteinander zu koppeln und somit Funktionalitäten zu bieten, die den Nutzer unterstützen.

3.3 Verwertungspotential

Die in ADDO entstandenen wissenschaftlichen Ergebnisse und die Implementation lassen sich hinsichtlich der wissenschaftlichen Weiterentwicklung und der industriellen Nutzung in Softwareprojekten wieder verwenden. Es wurde darauf geachtet, dass die Ergebnisse sowohl in Verbindung mit der entwickelten Semantik, als auch allein durch manuelle Konfiguration nutzbar sind. Die entstandenen Ontologien stellen nicht nur konzeptionell eine Weiterentwicklung von OWL-S dar, sondern auch das Schema ist direkt von OWL-S abgeleitet. Anderen Forschungsprojekten ergibt sich dadurch die Möglichkeit, bereits realisierte OWL-S Lösungen weiter zu verwenden. Auch das implementierte Framework bietet direkt Anknüpfungspunkte an eine API für OWL-S, ADDO Ontologien und das Dienstvermittlungssystem.

Das Dienstvermittlungssystem von ADDO erlaubt eine automatische Bindung von Diensten an Applikationen. Das automatische Binden ist dabei abhängig von semantischen Beschreibungen. Aber auch ohne semantische Beschreibungen ist der Dienstvermittler als Softwarekomponente für die industrielle Nutzung von Interesse, weil das ADDO Framework auf der Web Service Technologie basiert. Daher lässt sich das Binden von Applikationen und Diensten nicht nur durchführen, sondern auch managen. Neue Dienste lassen sich nahtlos in eine Umgebung integrieren und können sowohl ad-hoc als auch manuell auf Dienste abgebildet werden. Eine entsprechende Management-Applikation ist bereits implementiert und Teil des ADDO Frameworks.

4 Publikationen

4.1 Kongressbeiträge

An Ontology for Quality-Aware Service Discovery

Steffen Bleul und Thomas Weise

Tagungsband des First International Workshop on Engineering Service Compositions (WE-SC'05), Dezember 2005, Amsterdam, Holland.

Abstrakt: *The fast emergence and acceptance of service oriented architectures leads to fast development of extensional technologies like service delivery, discovery and composition. As main effort is being spent on automatic discovery and composition, current solutions do not reflect real world scenarios sufficiently. Services are offered by different vendors with different quality levels and prices. Large service oriented architectures with dynamic service compositions are not able to adapt without manual inspection of service quality and negotiation of service contracts. We propose an ontology for modelling Quality of Services (QoS) and Service- Level-Agreements (SLA). A semantic approach should bridge the gap of different terminology, languages and metrics making Service-Level offers and requests agent understandable and automatic quality-aware discovery possible.*

Automatic Quality-Aware Service Discovery and Matching

Steffen Bleul und Kurt Geihs

Tagungsband der 13th Annual Workshop of HP OpenView University Association (HP-OVUA), Mai 2006, Nizza, Frankreich.

Abstrakt: *Quality of Service (QoS) management provides a solid ground for integrating services in business processes. For example, reliability, fault tolerance and security are important factors to achieve a secure and fail safe execution of services. Therefore, quality of service is an essential factor for service discovery. Existing matching algorithms provide the service consumer a choice of services with the requested QoS followed by negotiation with the service vendor about a service level contract. We introduce a semantic model and framework for service level brokering where service vendors and consumers can specify their interests in service levels. Service level requests are matched against offers to discover the appropriate service. Service integration is done automatically by notifying service vendor and consumer about the integration. A semantic approach supports the system with matchmaking to improve the success of service discovery.*

Large-Scale Service Composition in Semantic Service Discovery.

Steffen Bleul, Thomas Weise und Kurt Geihs

Tagungsband der IEEE Joint Conference on E-Commerce Technology (CEC'06) and Enterprise Computing, E-Commerce and E-Services (EEE'06), Juli 2006, San Francisco, USA.

Abstrakt: *Self-Healing and self-optimizing service based applications are important steps towards the self-organizing Service Oriented Architectures (SOA). Self-Organizing SOAs replace services by functional equivalent services in the case of faults or in respect of quality of service. These features depend*

on automatic service discovery which provides service alternatives. We enter the WSC'06 contest to present a semantic service discovery system for large sets of services. A recursive algorithm builds service compositions by adding services in each iteration. The search works backwards, since we add services that produce a certain output regardless of its input parameters. A valid service composition produces a set of queried output parameters and input parameters necessary for the composed services. The algorithm is improved by using efficient data structures in our service composition system.

Ontology-Based Self-Organization in Service-Oriented Architectures

Steffen Bleul und Michael Zapf

Tagungsband des Workshop SAKS 2007, März 2007, Bern, Schweiz.

Abstrakt: *In Service-oriented architectures (SOA), services can be deployed and removed at runtime and can be arranged to business processes. Service-based applications and business processes can be reconfigured and optimized by replacing the participating services. In large environments, service processes should employ self-properties to improve manageability and flexibility. Using role descriptions, self-configuration becomes possible. We demonstrate an approach, realized in the ADDO project, which includes ontologies and QoS properties to achieve self-optimization within service processes.*

4.2 Sonstige Publikationen

An Ontology for Quality-Aware Service Discovery

Steffen Bleul, Thomas Weise und Kurt Geihs

Akzeptiert zur Publikation in der Spezialausgabe von Engineering Design and Composition of Service-Oriented Applications of Computer Systems Science Engineering (CSSE).

Abstrakt: *The fast emergence and acceptance of service oriented architectures leads to fast development of extensional technologies like service delivery, discovery and composition. As most effort is spent on automatic discovery and composition, current solutions do not reflect QoS negotiation sufficiently. Services are offered by different vendors with different quality levels and prices. Large service oriented architectures with dynamic service compositions are not able to adapt without manual inspection of service quality and negotiation of service contracts. We propose an ontology for modelling Quality of Services (QoS) and Service Level Agreements (SLA). Our semantic approach bridges the gap of different terminology, languages and metrics. Thus, Service Level Offers and requirements become machine understandable and automatic quality-aware discovery becomes possible.*

Flexible Automatic Service Brokering for SOAs

Steffen Bleul, Michael Zapf und Kurt Geihs

Akzeptiert zur Publikation im Tagungsband der 10th IFIP / IEEE Symposium on Integrated Management (IM 2007), Mai 2007, München, Deutschland.

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.

Making a Fast Semantic Service Composition System Faster

Steffen Bleul, Thomas Weise und Kurt Geihs

Akzeptiert zur Publikation im Tagungsband der der IEEE Joint Conference on E-Commerce Technology (CEC'07) and Enterprise Computing, E-Commerce and E-Services (EEE'07), Juli 2006, Tokyo, Japan.

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.*

4.3 Preise und Auszeichnungen

1. Platz bei Web Service Challenge 2006 (WSC206) in Semantischer Komposition von Diensten.

Es handelte sich um die dritte Veranstaltung dieses jährlich stattfindenden Wettbewerbs, die im Rahmen der IEEE Joint Conference on E-Commerce Technology (CEC'06) und Enterprise Computing, E-Commerce und E-Services (EEE'06) ausgetragen wurde. Zehn Teams aus China, Deutschland, Italien, Österreich, Singapur und den USA nahmen am Wettbewerb teil. Die in ADDO entwickelte Software war bei allen Aufgaben schneller als die Konkurrenz und gewann mit deutlichem Vorsprung.

Gegenstand des Wettbewerbs ist das Finden von passenden Dienstaggregaten zu einer Dienstanfrage eines Kunden. Dazu ist zum einen eine sehr große Menge unterschiedlichster Dienstangebote vorgegeben. Auf der Nachfragerseite werden Suchanfragen formuliert. Der Wettbewerb bestand nun darin, ein Programm zu entwickeln, das die Menge der Dienstangebote möglichst effizient und schnell durchsucht, um passende Dienste gemäß der vorgegebenen Suchanfrage ausfindig zu machen.

Large-Scale Service Composition in Semantic Service Discovery.

Steffen Bleul, Thomas Weis und Kurt Geihs.

IEEE Joint Conference on E-Commerce Technology (CEC'06) and Enterprise Computing, E-Commerce and E-Services (EEE'06), Juli 2006, San Francisco, USA.



5 Fazit

ADDO hat innerhalb der Projektlaufzeit erfolgreich Resultate für die automatische Dienstvermittlung geliefert und ist damit einen wichtigen Schritt hin zum Selbst-Management einer SOA gegangen. Das ADDO-Projekt realisiert ein umfangreiches Framework, bestehend aus Ontologie-Schemata, Abgleichalgorithmen zur Dienstsichtung und eine Architektur für automatische Dienstvermittlung. Das Framework ermöglicht erstmalig die automatische Sichtung und Bindung von Diensten an Dienstprozesse unter der Berücksichtigung von Dienstgüte. Die Ergebnisse wurden in zahlreichen Publikationen auf internationalen Konferenzen präsentiert. Das semantische Kompositionssystem hat auf einem internationalen Wettbewerb den ersten Platz belegt.

ADDO setzt sich klar von anderen Ansätzen internationaler Projekte ab. Die Ergebnisse sind in anderen Forschungsdisziplinen, z.B. Grid Computing und Ubiquitous Computing, in denen Dienste eine wichtige Rolle spielen, von großer Bedeutung. Sie stellen die notwendige solide Basis für das bereits genehmigte Nachfolgeprojekt ADDOaction dar. Innerhalb des Folgeprojektes wird das Selbst-Management von dienstorientierten Architekturen erforscht, um kostenpflichtige und vertragsgebundene Dienste automatisch zu integrieren und zu überwachen.

5.0.1 Literatur

- [1] Concierge OSGi.
URL: <http://concierge.sourceforge.net/>.
- [2] Eclipse Equinox an OSGi R4 Core Framework Specification.
URL: <http://www.eclipse.org/equinox/>.
- [3] Knopflerfish - Open Source OSGi.
URL: <http://www.knopflerfish.org/>.
- [4] Oscar - An OSGi framework implementation.
URL: <http://oscar.objectweb.org/>.
- [5] Web Service Challenge 2006.
URL: <http://www.ws-challenge.org/>.
- [6] Web Service Challenge 2007.
URL: <http://www.ws-challenge.org/>.
- [7] iDrive, iWalk: BMW takes Navigation to its logical Conclusion, November/December 2003.
URL: http://www.osgi.org/documents/news_events/osgi_tech_news/ATBBMW.pdf.
- [8] Luciano Baresi, Carlo Ghezzi, and Sam Guinea. Smart Monitors for Composed Services. In *ICSOC '04: Proceedings of the 2nd International Conference On Service Oriented Computing*, pages 193–202, New York, NY, USA, 2004. ACM Press.
- [9] Steffen Bleul and Kurt Geihs. Automatic Quality-Aware Service Discovery and Matching. In *Proceedings of the 13th Annual Workshop of HP OpenView University Association (HP-OVUA)*, pages 109–118, Kassel, Germany, May 2006. Infocomics-Consulting, Stuttgart, Germany.
- [10] Steffen Bleul and Thomas Weise. An Ontology for Quality-Aware Service Discovery. In *Proceedings of the First International Workshop on Engineering Service Compositions (WESC'05)*, pages 35–42, December 2005.
- [11] Steffen Bleul, Thomas Weise, and Kurt Geihs. An Ontology for Quality-Aware Service Discovery. Accepted for publication in the special issue on Engineering Design and Composition of Service-Oriented Applications, 2006.
- [12] Steffen Bleul, Thomas Weise, and Kurt Geihs. Large-Scale Service Composition in Semantic Service Discovery. In *IEEE Joint Conference on E-Commerce Technology(CEC ' 06)and Enterprise Computing, E-Commerce and E-Services(EEE ' 06)*, pages 427–429. Springer Press, June 2006.
- [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 and Michael Zapf. Ontology-based self-organization in service-oriented architectures. In *Proceedings of the Workshop SAKS 2007*, Bern, Swiss, March 2007. VDE.
- [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] Michael Blow, Yaron Goland, Matthias Kloppmann, Frank Leymann, Gerhard Pfau, Dieter Roller, and Michael Rowley. BPELJ: BPEL for Java. BEA and IBM, March 2004.
URL: <http://ftpna2.bea.com/pub/downloads/ws-bpelj.pdf>.
- [18] Mark Burstein, Jerry Hobbs, Ora Lassila, Drew Mcdermott, Sheila Mcilraith, Srinu Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, and Katia Sycara. OWL-S: Semantic Markup for Web Services. Website, November 2004.
URL: <http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/>.

- [19] M. R. Cabrer, R. P. D. Redondo, A. F. Vilas, J. J. P. Arias, and J. G. Duque. Controlling the smart home from tv. *Consumer Electronics, IEEE Transactions on*, 52(2):421–429, 2006.
- [20] Onyeka Ezenwoye and S. Masoud Sadjadi. RobustBPEL-2: Transparent Autonomization in Aggregate Web Services Using Dynamic Proxies. Technical Report FIU-SCIS-2006-06-01, School of Computing and Information Sciences, Florida International University, 11200 SW 8th St., Miami, FL 33199, June 2006.
- [21] Asuncion Gomez-Perez, Rafael Gonzales-Cabero, and Manuel Lama. ODE SWS: A Framework for Designing and Composing Semantic Web Services. 19(4):24–31, July, August 2004.
- [22] Information Society Technologies. *AMIGO: Ambient Intelligence for the Networked Home Environment*, 2004.
URL: <http://www.hitech-projects.com/euprojects/amigo/index.htm>.
- [23] Ryutaro Kawamura and Hiroyuki Maeomichi. Standardization Activity of OSGi (Open Services Gateway Initiative). *NTT Technical Review*, 2(1):94–97, 2004.
URL: <http://www.ntt.co.jp/tr/0401/files/ntr200401094.pdf>.
- [24] Heiko Ludwig, Alexander Keller, Asit Dan, Richard P. King, and Richard Franck. *Web Service Level Agreement (WSLA) Language Specification*. International Business Machines Corporation (IBM), August 2002.
URL: <http://www.research.ibm.com/wsla/WSLASpecV1-20030128.pdf>.
- [25] Ganna Frankova Marco Aiello and Daniela Malfatti. What’s in an Agreement? An Analysis and an Extension of WS-Agreement. Proceedings of the International Conference on Service-Oriented Computing (ICSOC), 2005.
- [26] Abhijit Patil, Swapna Oundhakar, Amit Sheth, and Kunal Verma. METEOR-S Web Service Annotation Framework. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW2004)*, New York, USA, May 2004. LSDIS Lab, University of Georgia.
- [27] Akhil Sahai, Anna Durante, and Vijay Machiraju. *Towards Automated SLA Management for Web Services*. Hewlett-Packard Laboratories Palo Alto, July 2002.
URL: <http://www.hpl.hp.com/techreports/2001/HPL-2001-310R1.pdf>.
- [28] Robbie Schäfer, Steffen Bleul, and Wolfgang Müller. Dialog Modelling for Multiple Devices and Multiple Interaction Modalities. In *2006 Workshop on Task Models and Diagrams for UI Design (TAMODIA 2006)*, Hasselt, Belgium, October 2006.