

C. Zirpins und W. Lamersdorf

# Dienstorientierte Kooperationsmuster in servicebasierten Grids

## ZUSAMMENFASSUNG

Dienstorientierte Informationssysteme zielen auf die gleichzeitige Steigerung von Effizienz und Effektivität interorganisatorischer Kooperation. Der Dienstbegriff dieser Ebene ist eng mit dem generellen Prozessbegriff und speziell mit Workflowkonzepten verbunden. Prozesse erfassen die Koordinationslogik für eine Vielzahl von Ressourcen, mit denen sich wechselnde Teilnehmer in temporäre Kooperationsbeziehungen einbringen. Die systematische Unterstützung des komplexen Gesamtvorgangs kann durch ganzheitliche Strukturierung in geordnete Kooperationsmuster wesentlich erleichtert werden. In diesem Beitrag wird ein Rahmenwerk zur Realisierung dienstorientierter Kooperationsmuster vorgestellt, das im Kontext des FRESCO (Foundational Research on Service Composition) Projektes entstanden ist. Der Ansatz schlägt ein konzeptuelles Dienstmodell sowie ein konkretes workflowbasiertes Architekturmodell vor, das auf dem von der WfMC (Workflow Management Coalition) vorgeschlagenen Standard Workflow Metamodel aufsetzt. Ein Managementkonzept skizziert grundlegende Funktionen für Planung, Konstruktion und Betrieb dienstorientierter Lösungen. Der technische Teil des Rahmenwerks bildet eine Plattform zur Realisierung von Diensten auf Basis von OGSA (Open Grid Service Architecture), Gridtechnologien und BPEL (Business Process Execution Language) Prozesstechnologien.

## 1 EINLEITUNG

*Kooperationsmuster* sind durch wiederkehrende Interaktionsprozesse zwischen wechselnden Partnern charakterisiert und in zahlreichen Anwendungsgebieten wie E-Business, E-Government oder E-Learning zu finden. Zwischenbetriebliche Integrationsprobleme [1], die auf das dynamische Verhältnis zwischen Anbietern und ihren regelmäßig wechselnden Partnern abzielen, liefern typische Beispiele: In Situationen wie der flexiblen Auslagerung von Geschäftsfunktionen (*Outsourcing*) oder der dynamischen Organisation von Zulieferketten (*Supply Chain Management*) sehen sich die beteiligten Unternehmen ähnlichen, immer wiederkehrenden Kooperationsformen mit jeweils wechselnder Besetzung gegenübergestellt.

Sind Kooperationsmuster durch einseitige Leistungserbringung konstanter Leistungserbringer zu Gunsten wechselnder Leistungsnehmer charakterisiert, spricht man von *Dienstleistungen*. Dienstleistungen basieren auf einer Vielzahl von Prozessen. Eine erste Kategorisierung dieser Dienstprozesse kann auf Basis von internen Erstellungs- und externen Erbringungsprozessen erfolgen. Erstellungsprozesse umfassen die Fähigkeiten eines Dienstbringers, die notwendig sind, um die Inhalte seiner Leistung zu realisieren. Beispielsweise gehören Handhabung und Transport von Gütern zu den notwendigen Fähigkeiten im logistischen Umfeld. Erbringungsprozesse umfassen hingegen all die Fähigkeiten des Dienstbringers, mit denen er Kunden die Dienstinhalte verfügbar und nutzbar macht. Im

Logistikbeispiel gehören dazu etwa die Veröffentlichung von Lieferzielen und -kapazitäten, Vertragsverhandlungen und Kundeninteraktion. Dienstprozesse in und zwischen Kategorien sind dabei eng miteinander verknüpft. Zum Beispiel führt die zusätzliche Erbringung von Auftragsverfolgung zur notwendigen Erweiterung der Güterhandhabung durch einen zusätzlichen Erfassungsvorgang.

Die Unterstützung derartig komplexer Kooperationsmuster erfordert eine systematische Modellierung. Diese muss zudem in einer Weise repräsentiert werden, die gleichsam von den menschlichen Entwicklern leicht verstanden und von der Ausführungsumgebung effizient verarbeitet werden kann.

### 1.1 Der fundamentale Dienstbegriff

Im weiteren Verlauf soll ein Dienstbegriff entwickelt werden, der das beschriebene Bild dienstorientierter Kooperationsmuster fasst. Dabei werden zunächst die klassischen, aus offenen verteilten Systemen bekannten Konzepte zu Grunde gelegt und entsprechend erweitert.

Das allgemeine technische Verständnis ist am nachhaltigsten vom ODP Standard [2] geprägt worden, der daher als Basis dienen soll, um den hier anvisierten erweiterten Dienstbegriff zu konkretisieren. Grundsätzlich wird eine konstante Klasse von Kooperationen (*Diensttyp*) von den daraus abgeleiteten konkreten Kooperationsfällen (*Dienstinstanzen*) unterschieden. Dienstinstanzen variieren dabei in Bezug auf die Konditionen der Kooperation, die als Dienstigenschaften (etwa QoS) bezeichnet werden und sich aus den spezifischen Merkmalen der aktuellen Teilnehmer ableiten. Diese Teilnehmer werden durch *Rollen* abstrahiert, welche im Zuge der Dienstbeziehung ein erwartetes Kooperationsverhalten bezeichnen. Von der *Providerrolle* wird dabei erwartet, Typ und Eigenschaften von Instanzen offen zu legen, für die eine Beteiligung erwünscht ist, und Kooperationspartner gesucht werden. Die *Klientenrolle* beobachtet das Angebot (eines Marktes) in Bezug auf einen spezifischen Typ, wählt einen Provider in Bezug auf dessen Dienstigenschaften und geht mit diesem eine Kooperationsbeziehung ein. Spezifische Interdependenzen zwischen Diensten werden oft als *Dienstkomposition* bezeichnet. In diesem Fall setzt ein Teilnehmer einen *komplexen Dienst* (oder *Verbunddienst*), für den er als Provider agiert, mit *Dienstkomponenten* in Verbindung, bei denen er als Klient mitwirkt und bestimmt, in welcher Weise sich die Charakteristiken des Verbunddienstes aus denen der Dienstkomponenten herleiten.

Bezüglich ihrer Charakteristik weisen Dienstleistungen auf Anwendungsebene (z.B. Transportdienste), wie oben gesehen, einen weitaus höheren Komplexitätsgrad auf als die technisch orientierten Services klassischer Szenarien (z.B. ein Druckdienst). Neben der schwierig zu erfassenden konkreten Semantik des Kooperationsziels (z.B. Hochsicherheits-Gefahrgut-Express-Transfer) ist es vor allem der „syntaktische“ hochkomplexe Kooperationsprozess, der die

wesentliche Charakteristik eines organisatorischen Dienstes ausmacht. Der Fokus liegt hier insbesondere auf den zu beobachtenden Interaktionen, d.h. Kommunikationsprozessen zwischen Rollen.

Das Problemfeld, dem sich Organisationen in Bezug auf die Teilnahme an dienstorientierten Kooperationsbeziehungen gegenübergestellt sehen, kann in *strategische*, *taktische* und *operationale* Herausforderungen gegliedert werden. Auf strategischer Ebene liegt das Augenmerk zunächst auf der Erkennung von Dienstypen und ihren wechselseitigen Abhängigkeiten. Diese müssen im Zuge der *Dienstmodellierung* explizit gemacht werden, wozu ausdrucksstarke Modelle und systematische Entwurfsmethodologie Voraussetzung sind, die sowohl den technischen als auch konzeptionellen Kontext der Teilnehmer einbeziehen.

Auf taktischer Ebene müssen Dienstypen im Rahmen von *Diensttypadaption* konstant gepflegt werden, um mit organisatorischen Veränderungen Schritt zu halten, die auf Grund der Abhängigkeit zu einem fluktuierenden Kontext (etwa einem Wirtschaftsmarkt) unvorhergesehen eintreten können und schnelle Reaktionsfähigkeit fordern. Des Weiteren müssen auf dieser Ebene potentielle Partner für Dienste lokalisiert werden. Je nachdem ob eine Organisation als Provider oder Klient auftreten möchte, ist dazu eine Veröffentlichung (*Publication*) oder Suche (*Discovery*) nötig.

Auf operationaler Ebene müssen die Partner dann mit Dienstypen abgeglichen (durch den Provider im Falle von Klienten) und ausgewählt (durch den Klienten im Falle von Providern) werden, was als *Dienstagggregation* bezeichnet wird. Bei Dienstverbunden muss ein Provider zusätzlich eine *Dienstkomposition* durchführen, wobei weitere Provider für Dienstkomponenten unter der Bedingung auszuwählen sind, dass die Anforderungen von Klienten des Dienstverbundes erfüllt werden. Während der eigentlichen Dienstinteraktion müssen die Bedingungen und Konditionen des Dienstes durch *Dienstkoordination und -kontrolle* sichergestellt werden. Zusätzliche Flexibilität kann dabei noch durch dynamische Änderungen von Dienstinstanzen (*Dienstinstanzadaption*) erreicht werden, was aber mit hohem Aufwand einhergeht.

Über alle Ebenen hinweg sind übergreifende Systemmechanismen notwendig, welche die üblichen diversifizierten Umgebungen organisatorischer Informations- und Kommunikationstechnik im Sinne *kooperativer Informationssysteme* [3] organisieren, um damit zum einen Dienste realisieren zu können und zum anderen Unterstützung für die oben genannten Aufgaben zu erhalten. Eine derartige Middleware soll als *Dienstmanagementsystem (DMS)* bezeichnet werden. Des weiteren werden die Aufgaben der Planung, der Realisierung und des Betriebens von dienstorientierten verteilten Anwendungen als *Dienstmanagement* zusammengefasst.

## 1.2 Stand der Technik

Aktuelle Techniken dienstorientierter Systeme (bzw. *Service Oriented Computing*) sind bisher primär auf technische Aspekte ausgerichtet [4]. Die Unterstützung von Dienstleistungen der Anwendungsebene geht hingegen über die Reichweite der meisten Ansätze hinaus. Auf der anderen Seite ebnet sie aber mit wesentlichen Grundfunktionen den Weg dorthin.

Vorneweg verspricht der aufstrebende Webservice Standard [5, 6] die Interoperabilität quasi beliebiger heterogener Systemlandschaften. Dazu wird die Ausdrucksmächtigkeit von XML zunächst zur Spezifikation operationaler Interfaces verwandt und dient zwecks deren Zugriffs des Weiteren zur Realisierung von Kommunikationsprotokollen, die über offene Internetkommunikation abgewickelt werden können. Dies erlaubt es Organisationen, ihre internen Informationssysteme in geregelter Weise als webbasierte Komponenten offen zu legen. Ansätze des *Grid Computing* [7] gehen noch einen Schritt weiter, indem sie Webservice-Techniken benutzen, um den gesamten (langen) Lebenszyklus beliebiger Ressourcen (*Grid Services*) ähnlich Geschäftsobjekten zu verwalten und sie in sog. *Service Grids* zu strukturieren, über die sie dann einheitlich und geregelt zugreifbar sind.

Auf organisatorischer Ebene realisieren Servicekomponenten Endpunkte (oder *Ports*) für Interaktionen, über die eine Teilnahme an automatisierten Kooperationen zwischen verschiedenen Organisationen möglich wird. Bezüglich der Kooperationsprozedur wird durch das der Webservice Architektur zugrunde liegende *Service Oriented Model (SOM)* nur ein sehr grundlegendes Interaktionsmuster definiert, das als *Brokerdreieck* allgemein bekannt ist. Darüber hinaus erlauben webservicebasierte (*Work*)*Flow* Sprachen wie BPEL4WS [8] die Definition einfacher individueller Interaktionsprozesse, die jedoch wiederum komplett von dienstorientierten Aspekten abstrahieren. Dies markiert den Übergang zur aktuellen Forschung in verschiedenen Bereichen kooperativer, zwischenbetrieblicher Geschäftsprozesse, Workflow und Servicekomposition.

## 1.3 Das FRESCO Projekt

Das FRESCO Projekt beschäftigt sich mit der Konzeption und Realisierung eines grundlegenden dienstorientierten Kooperationsansatzes [9]. Projektziel ist die Entwicklung eines konzeptionellen und technischen Rahmenwerks, das Organisationen dabei unterstützt, eigene Dienste als multilaterale Kooperationsbeziehungen zu realisieren. Im Zuge dessen wurde ein fundamentales Dienstmodell entwickelt, das Anwendungsdienste als kooperative Interaktionsmuster beschreibt. Dieses Modell wurde im Rahmen eines generischen Dienstmanagementsystems auf Basis von Web- und Gridservice-Technologien implementiert.

Im Weiteren wird der FRESCO Ansatz im Detail beschrieben. Der zweite Abschnitt dieses Artikels stellt das konzeptionelle Dienstmodell ein aufbauendes Architekturmodell sowie ein Konzept zum Management derartiger Dienste vor. Anschließend wird in Abschnitt drei die praktische Realisierung der Konzepte durch das FRESCO Toolkit skizziert. Die letzten beiden Abschnitte beinhalten eine Diskussion verwandter Ansätze sowie ein Fazit.

## 2 EIN KONZEPTIONELLER RAHMEN FÜR DIENSTE

Das FRESCO Rahmenwerk basiert auf einem konzeptionellen Modell, das Anwendungsdienste als strukturierte Mengen kooperativer Interaktionsprozeduren (Kooperationsmuster) definiert. Diese Konzepte implizieren ein spezifisches Architekturmodell, das auf einem homogenen Komponentenmodell und Workflow basiert. Auf Basis dieser Architektur werden essentielle Funktionen zum Planen, Konstruieren und Betreiben von dienstorientierten Anwendungen als Systemkomponenten konzipiert, die eine komplette zusammenhängende DMS Umgebung ergeben.

### 2.1 Konzeptionelles Dienstmodell

Das konzeptionelle Dienstmodell [10] definiert eine betreiberorientierte und dienstzentrierte Sicht auf Anwendungsdienste, wobei der Schwerpunkt auf Kooperationsmustern liegt, die sich aus den Interaktionsprozessen der beteiligten Partner ableiten. Kooperationsprozeduren, die atomare, abgeschlossene Teile einer Dienstbeziehung darstellen, werden als so genannte *Dienstfähigkeiten* hervorgehoben. Genauer repräsentieren Dienstfähigkeiten *Zweck*, *Interaktionslogik* sowie *Resultat* einer Teilkooperation zwischen organisatorischen *Rollen*. Dabei werden durch Dienstfähigkeiten zusätzliche koordinative Rollen eingeführt, die eine Indirektionsebene zwischen den eigentlich interagierenden Rollen darstellen. Dabei sind Dienstfähigkeiten als vollwertige Teilnehmerrollen erster Klasse zu verstehen, die ihre Interaktionsmuster aktiv durchsetzen. Ein Dienst ist durch die Menge seiner Fähigkeiten gekennzeichnet.

Ein wesentliches Merkmal des Modells ist die Unterscheidung von Dienstfähigkeiten in Bezug auf *Diensterstellung* und *Diensterbringung*. Diensterstellung bezieht sich auf die Leistung eines Dienstes (z.B. Gütertransfer). Es wird angenommen, dass sich diese Leistung aus spezifischen Ressourcen des Providers herleitet (z.B. Geschäftsprozesse, Know-how, Humankapital, Maschinen, usw.). Um die Diensterstellung zu erfassen, werden Prozesse, die Interaktionen mit derartigen Ressourcen beinhalten, explizit als bedeutsame Leistungsmerkmale (z.B. Auftragsverfolgung) hervorgehoben und entsprechende Fähigkeiten als *Erstellungsfähigkeiten* bezeichnet. Erstellungsprozeduren beinhalten keine weiteren Interaktionen zwischen Rollen, sondern sind Monologe der Providerrolle zum Einbezug von Ressourcen (ähnlich *Bindings* in [11]), die den Klienten indirekt durch andere Fähigkeiten *erbracht* werden müssen. Die grundlegenden Erstellungsfähigkeiten eines Dienstes werden als *Dienstleistungskern* zusammengefasst, der das gesamte Leistungsspektrum repräsentiert.

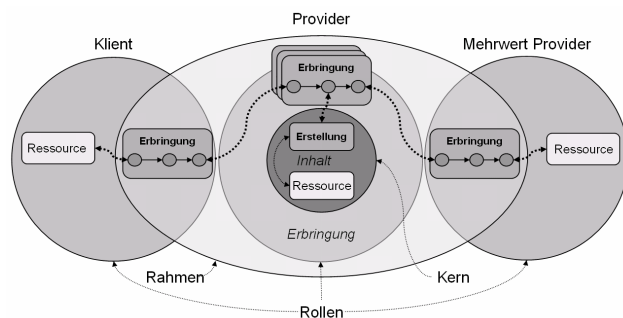


Abb.1 Konzeptionelles Dienstmodell

*Diensterbringung (Provision)* bezeichnet hingegen diejenigen Fähigkeiten, die einen Dienst antreiben und seine Inhalte verfügbar machen (z.B. Verhandlung von Bedingungen und Konditionen, Einbezug von Anlagefähigkeiten, Leistungsabrechnung). Diensterbringungsfähigkeiten verfügen über die exklusive Kontrolle aller Interaktionsprozeduren und üben diese proaktiv aus. Die Menge aller Erbringungsfähigkeiten ist in einer Schicht um die Erstellungsfähigkeiten des Kerns gruppiert, die als Dienstleistungsrahmen (*Service Shell*) bezeichnet wird.

Innerhalb des Dienstleistungsrahmens sind die Erbringerfähigkeiten wechselseitig miteinander verknüpft und teilen eine gemeinsame Sicht auf Rollen und erbringungsrelevante Informationen. Verknüpfungen begründen das Gesamtverhalten der Diensterbringung, indem sie das globale Zusammenspiel von Fähigkeiten definieren. Eine komplette

Dienstleistungsbeschreibung beinhaltet die Definition eines grundlegenden Kerns sowie vor allem des interaktionsbestimmenden Rahmens (Abb.1). Unser Fokus liegt primär auf letzterem.

### 2.2 Workflowbasiertes Architekturmodell

Der konzeptionelle Dienstbegriff wird durch ein Architekturmodell konkretisiert, das als dienstorientierte Architektur (DOA) bezeichnet wird. DOA nutzt Workflowkonzepte, um Dienste als partitionierte Menge zusammenhängender Komponenten mit präzisiertem Interaktionsverhalten zu definieren, wobei eine Teilmenge C (Ressourcen) interagierende Teilnehmer darstellt und eine Teilmenge S (Fähigkeiten) deren Interaktionsmuster repräsentiert und umsetzt.

Im Detail definiert SOA einen Dienst durch ein Kooperationsmuster, das sich aus den *Fähigkeiten* zur Kontrolle von *Interaktionsmustern* zwischen den *Ressourcen* beteiligter *Rollen* ergibt. Eine Rolle  $r \in R$  repräsentiert die Verantwortung für eine Menge von Ressourcen, die durch Komponenten  $C_r$  repräsentiert werden und für die Teilnahme an einer Dienstbeziehung zwingend notwendig sind. Konkret werden Komponenten durch ihre Kommunikationsendpunkte  $E_x = \cup_{c \in C_r} c$  charakterisiert, die atomare Interaktionen ermöglichen. Während einer Interaktion werden Datenartefakte  $D_e \in D$  kommuniziert. Fähigkeiten bilden Bindeglieder zwischen Komponenten und repräsentieren eine in sich geschlossene Kooperationsprozedur.

Der Dienst- oder Kooperationsrahmen ist durch eine Menge von Fähigkeiten  $S \subset C \times R \times P^*$  gegeben, wobei jedes  $s = (c, r, P_s)$  eine Komponente repräsentiert, die eine Menge von Interaktionsprozessen durchsetzt und explizit einer Rolle zugeordnet ist. Ein Prozess  $p \in P_s$  definiert eine Menge von Transitionen, die einen Präzedenzgraphen zwischen Interaktionsaktivitäten  $A_p \subset E_c \times R \times D^* \times \{in, out\}$  ergeben. Für die Fähigkeit  $s = (c_s, r_s, P_s)$  repräsentiert eine Aktivität  $a_1 = (e_{c_s}, r_s, D_1, in)$  eine eingehende Interaktion, die extern herbeigeführt wurde und die Kommunikation von Datenartefakten  $D_1$  mit dem – von der die Fähigkeit erbringenden Rolle  $r_s$  selber bereitgestellten – Endpunkt  $e_{c_s}$  beinhaltet. Für die gleiche Fähigkeit repräsentiert eine weitere Aktivität  $a_2 = (e_x, r_y, D_2, out)$  eine ausgehende Interaktion, bei der  $r_s$  die Kommunikation von Datenartefakten  $D_2$  selber herbeigeführt und der Endpunkt  $e_x$  von einer anderen Rolle  $r_y$  bereitgestellt wird. Für zwei Fähigkeiten  $s = (c_s, r_s, P_s)$  und  $t = (c_t, r_t, P_t)$  mit darin enthaltenen Interaktionsprozessen  $p_1 \in P_s$  und  $p_2 \in P_t$ , definieren die Aktivitäten  $a_1 = (e_{c_t}, r_t, D, out)$ ,  $a_2 = (e_{c_t}, r_t, D, in)$  eine *Komposition*, wobei  $a_1$  eine ausgehende Interaktion von  $p_1$  und  $a_2$  eine eingehende Interaktion von  $p_2$  ist.

Der so definierte Begriff der Fähigkeit konkretisiert das Fähigkeitskonzept des Dienstmodells in dem Sinne, dass implizite Ontologieassoziationen der Architekturelemente die Semantik von Interaktionslogik definieren, die sich aus dem Fluss der Interaktionsaktivitäten ergibt und im Fluss von Datenartefakten resultiert. Darüber hinaus wird das Konzept der Fähigkeitsverknüpfung auf die Komposition derer Interaktionsprozesse abgebildet.

Da das DOA Modell sich auf Workflowkonzepten gründet, lässt es sich auf das Metamodell der Workflow Management Coalition (WfMC) [12] abbilden. Diensttypen werden mit Hilfe der Workflow Sprache XPDLL spezifiziert und im weiteren als *Diensteschemata* bezeichnet.

### 2.3 Dienstplattform

Das Dienstmanagementkonzept basiert auf einer modularen, erweiterbaren Managementumgebung, in der zunächst eine Menge grundlegender Basismechanismen realisiert ist. Neben Modellierung liegt der Fokus dabei vor allem auf Adaption, Aggregation und Koordination.

Da die ganzheitliche Realisierung von Dienstleistungen selbst in relativ einfach anmutenden Fällen zu inhärent komplexen Spezifikationen führt, ist für die Unterstützung ihres Entwurfs *Dienstmodellierungsunterstützung* notwendig. Eine grafische Modellierungssprache mit entsprechenden Werkzeugen sowie automatisierte Validierungsmöglichkeiten sind Voraussetzung, um Entwickler bei der Erstellung von Dienstschemata zu unterstützen. Dies ist zugleich der initiale Schritt eines fundamentalen Dienstleistungslebenszyklus und wird in der Regel durch den Provider erbracht.

*Dienstschemamanagement* erbringt in der Folge die notwendige Funktionalität zur programmatischen Verarbeitung der Schemata. Neben grundlegenden Repositoryfunktionen zur Speicherung und Lokalisierung liegen die wesentlichen Anforderungen in der systematischen und konsistenten Adaption. Hier legt die Komplexität der Schemata wiederum automatisierte Methoden nahe, wofür sich etwa regelbasierte Ansätze eignen, die eine präzise systematische Änderung großer Mengen von Interaktionsprozessen erlauben. Im Rahmen des Dienstlebenszyklus kann das Schema an dieser Stelle einer kontinuierlichen Anpassung und Weiterentwicklung durch den Provider unterzogen werden, bis es schließlich zur Anwendung kommt.

Dann ist es die Aufgabe der *Dienstaggregation*, eine Instanz des Dienstschemas zu erzeugen, der eine Zuordnung der organisatorischen Rollen zu konkreten Teilnehmern zugrunde liegt. Das Hauptproblem ist es, Ressourcen der Teilnehmer zu allozieren, die sich aus den ihrer Rolle im Schema zugeordneten Komponenten ableiten. Dabei soll zum einen die Allokation optimiert werden und zum anderen ein konstanter Ablauf der Dienstleistungsprozedur gewährleistet sein. Im schwierigsten Fall müssen hier wechselnde Teilnehmer oder auch Änderungen im Dienstleistungsschema berücksichtigt werden. Anfänglich muss zumindest der Provider bekannt sein, bei dem Ressourcen für eine initiale Dienstleistungsfähigkeit alloziiert werden können.

Dienstleistungsfähigkeiten werden durch Komponenten realisiert, deren wesentliche Aufgabe in der Koordination der den Fähigkeiten innewohnenden Teilkoooperationsmuster liegt und die als *Dienstengines* bezeichnet werden. Das Hauptproblem liegt hier für die verantwortlichen Teilnehmer darin, bei der Realisierung der ihnen zugeordneten Fähigkeiten den gesamten Dienstkontext inklusive möglicher Assoziation zu weiteren Fähigkeiten in anderen Dienstengines sowie das einheitliche Rollen- und Datenmodell zu berücksichtigen. In FRESCO wird dazu ein technisches Rahmenwerk vorgeschlagen, das mit interpretierbaren Spezifikationen parametrisiert wird, die aus dem Schema generiert werden. Der Dienstlebenszyklus geht hier in seine (inter-)aktive Phase. Wenn im Verlauf der Ausführung der Dienstleistungsinstanz schließlich alle Dienstengines (respektive die von ihnen realisierten Fähigkeiten) einen Endzustand erreicht haben, terminiert die Instanz, und der Dienstlebenszyklus fährt mit einer weiteren Runde statischer Schemaadaption fort.

Abb.2 zeigt einen Überblick der dienstorientierten Architektur mitsamt den Dienstmanagementmechanismen und ihren Zusammenhängen.

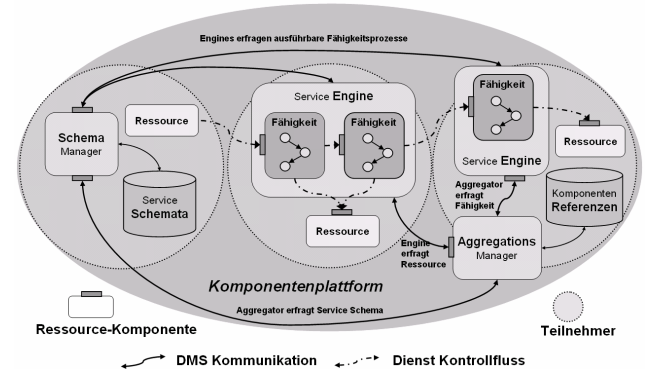


Abb.2 DOA mit DMS Mechanismen

Eine wesentliche Charakteristik dieser DMS Umgebung liegt in der Tatsache, dass alle Managementmechanismen wiederum selber Komponenten erster Klasse darstellen. Auf diese Weise können Änderungen von Dienstleistungen zur Erbringungszeit erfolgen und von den Dienstinstanzen selber angestoßen werden. Zum Beispiel kann eine Fähigkeit zur dynamischen Änderung von Teilnehmern führen (etwa durch Einführung neuer Teilnehmer im Rahmen einer Brokerfähigkeit) oder auch eine dynamische Adaption des Schemas hervorrufen (etwa bei Änderungen einer Abrechnungsfähigkeit als Resultat einer Verhandlungsfähigkeit). Es ist erwähnenswert, dass diese Charakteristik auch die Erweiterung der Managementmechanismen erlaubt, indem diese als Dienstleistungsfähigkeit realisiert werden.

### 3 FRESCO TOOLKIT

Als Konzeptnachweis wurden DOA und DMS Kernfunktionen im Rahmen des prototypischen *FRESCO Toolkit* implementiert<sup>1</sup> (Abb.3 zeigt die Architektur). Das Toolkit realisiert einen FRESCO-Dienst als Menge von Gridservice Komponenten [13], die in zwei Teilmengen unterteilt sind: Fähigkeiten und Ressourcen. *Ressourcen* sind als konventionelle Gridservices implementiert, die eine Dienstspezifische Erweiterung aus OGSA Servicedata Attributen realisieren. Fähigkeiten sind Komponenten, die ebenfalls als Gridservices zugegriffen werden, aber gleichzeitig in proaktiver Weise dienstspezifische Interaktionsprozesse umsetzen. Während die Kernfunktionen von Ressourcen außerhalb des Toolkits zu realisieren sind, werden die Fähigkeiten komplett durch DSM Mechanismen auf der Basis von Dienstschemaspezifikationen generiert und installiert. Intern kapseln Fähigkeiten eine BPEL Engine, die die Interaktionsprozesse antreibt. Da BPEL auf Webservice Technologie basiert, sind entsprechende Engines nicht grundsätzlich in einer Grid-Umgebung anwendbar. Daher beinhalten Fähigkeiten eine Integrationsarchitektur zwischen Grid- und Webservices. Die Integration wird dabei durch verschiedene Interzeptoren (genauer: Proxies und Adapter) erreicht, die Aufrufe transformieren und Informationen zum Instanzmanagement konvertieren.

Das Management von FRESCO-Diensten basiert auf einer Reihe von *Managementkomponenten*. Auf der einen Seite sind die Managementkomponenten Tools, die eine grafi-

<sup>1</sup> Das FRESCO Toolkit steht unter Academic Free Licence und kann frei bezogen werden ([www.servicecomposition.org](http://www.servicecomposition.org)).

sche Benutzerschnittstelle bieten. Auf der anderen Seite sind sie systemtechnische Unterstützungsmechanismen, die eine programmatische Schnittstelle zur Verfügung stellen. Die Komponenten sind grundsätzlich als allein stehende Java-Anwendungen implementiert, die jedoch intensiv interagieren und auch als Ressourcen von Diensten nutzbar sind. Daher sind die wesentlichen Managementfunktionen über das Grid ansprechbar. Jedes Tool besitzt einen Gridproxy, zu dem es über Java RMI Verbindung hält. Zusätzlich ist ein Repository für FRESKO-Dienstschemata als Webanwendung auf Basis von Servlets realisiert.

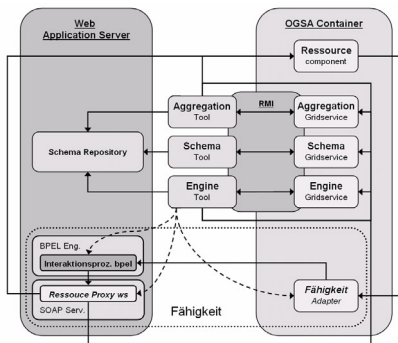


Abb. 3 FRESKO Toolkit Architektur

Im Folgenden werden die Kernfunktionen der Managementkomponenten skizziert, die sich grob in Schema- und Instanzmanagement gliedern lassen.

### 3.1 Schemamanagement

Die FRESKO Schemamanagementkomponente verwaltet die generischen Spezifikationen aller Dienstschemata und macht sie programmatisch zugreifbar. Ihre Hauptfunktion liegt jedoch in der Möglichkeit beliebige Transformationen auf die Schemata anzuwenden, welche eine kontrollierte Änderung der Struktur sowie Interaktionslogik von Diensten erlauben. Darüber hinaus ist es möglich, die Repräsentation von Interaktionsmustern in ein ausführbares Format zu transformieren. Evolutionäre Adaption von Dienstprozessen wird durch eine Sprache für Änderungsregeln unterstützt, die durch einen Regelinterpreter im Schemamanager durchgesetzt werden. Dieser erlaubt es, beliebige Muster in den XPD L Workflowprozessen der Dienstschemata zu selektieren und diese Muster dann zu entfernen oder durch neu erzeugte Workflowelemente zu ersetzen, wobei neue Revisionen der Schemata entstehen.

Schließlich müssen – zwecks Erzeugung von Dienstinstanzen – die Interaktionsmuster von Fähigkeiten, die in XPD L gegeben sind, in ein Format transformiert werden, das von Dienstengines ausführbar ist. Letztere realisieren damit aktive Komponenten, die Dienstleistungsfähigkeiten zur Laufzeit erbringen. Die Transformation basiert auf der Tatsache, dass die meisten Workflowsprachen einen gemeinsamen Satz von Kernkonzepten mit vergleichbarer Semantik teilen (siehe [14, 15]). In FRESKO wurde der aufstrebende BPEL4WS Standard als Ausführungsformat gewählt und eine Abbildung von XPD L definiert. Eine umfassende Betrachtung ist in der Dokumentation des FRESKO Toolkit zu finden.

### 3.2 Instanzmanagement

Das Management von Dienstinstanzen beinhaltet zunächst die Zuordnung von Teilnehmern, die eine konkrete Kooperationsbeziehung eingehen wollen. Von diesen werden

später Ressourcen alloziert, die eine Beteiligung an Interaktionsprozessen ermöglichen. Die FRESKO *Aggregator Komponente* wertet dazu Dienstschemata aus und ermittelt die darin enthaltenen Rollen mitsamt zugehörigen Ressourcen. Im Zuge des Ablaufs einer Dienstinstanz werden alle Rollen an Teilnehmer zugewiesen. Jeder Teilnehmer stellt die Komponenten zur Verfügung, welche die mit seiner Rolle assoziierten Ressourcen repräsentieren. Dabei kann eine individuelle Strategie bestimmt werden, die Einfluss darauf hat, wann (zeitlich) und wie (methodisch) Rollen zuordnung und Ressourcenallokation durchgeführt werden. Die im Schema enthaltenen Fähigkeiten werden speziell behandelt. Ermittelt der Aggregator die Zuordnung einer Rolle zu einer Fähigkeit, fordert er den entsprechenden Teilnehmer zu deren automatisierten Realisierung auf.

Die automatische Realisierung von Komponenten zur Ausübung von Dienstfähigkeiten geschieht dabei über eine Managementkomponente für Dienstengines (*Engine Manager*). Diese beantragt zunächst beim Schemamanager die Transformation der im Schema spezifizierten Interaktionsprozesse in das ausführbare BPEL Format. Daraufhin werden die notwendigen Interzeptoren generiert, mit denen zum einen die BPEL Prozesse aus dem Grid angesprochen werden können (Fähigkeitsadapter) und zum anderen der Zugriff von Gridservices durch eine BPEL Engine möglich wird (Ressourcenproxy). Hierbei sind grundsätzliche Unterschiede zwischen den eigentlich verwandten Technologiefamilien der Web- und Gridservices zu überbrücken. Die maßgebliche Differenz besteht im unterschiedlichen Umgang mit zustandsbehafteten Komponenteninstanzen, die im Grid durch robuste Zeiger und in BPEL durch Korrelation in Bezug auf Anwendungsdaten referenziert werden. Der Informationsverlust, der durch den Übergang zwischen langlebigen Gridservices und zustandslosen Webservices entsteht, geschieht mit Hilfe der Aggregationskomponente, die zur Abfrage individueller Komponentenreferenzen verwendet wird.

## 4 VERWANDTE ANSÄTZE

Die Untersuchung unternehmensübergreifender Prozesse ist Gegenstand einer Reihe von Forschungsbereichen mit verschiedenen Perspektiven und Schwerpunkten. (*Geschäfts-)Prozessmodellierung und -management* befasst sich mit Formen organisatorischer Abläufe und deren Organisation, wobei auch die Ausweitung über Organisationsgrenzen hinweg betrachtet wird. Hier werden speziell Modelle *virtueller Konzerne* und der sie verbindenden *Interkonzernprozesse* untersucht [16-22]. Das Projekt *DySCo* [23] führt einen elementaren prozessbasierten Dienstbegriff ein und entwickelt Methoden zur Ableitung teilnehmerspezifischer Anteile des Gesamtprozesses. In FRESKO dienen Prozessmodelle als Grundlage für die aufbauende konzeptionelle Modellierung von Diensten.

*Workflowmanagement* [24] untersucht die Unterstützung von (Geschäfts-)Prozessen durch Abbildung in hochgradig automatisierbare Arbeitsabläufe, wobei auch verteilte Abläufe einbezogen werden. Dabei wird der grundlegenden Strukturierung verteilter Abläufe [25] in Hinblick auf deren ganzheitliche Kontrolle sowie der technischen Realisierung verteilter Workflowmanagementsysteme [26-28] nachgegangen. Das FRESKO Architekturmodell bedient sich dem WfMC Workflow Metamodell [12], um konzeptionelle Dienste im Hinblick auf ihre Realisierung zu beschreiben. Die FRESKO Dienstplattform nutzt Workflow Verteilungskonzepte und wendet sie auf aktuelle servicebasierte Techniken an.

Im Teilbereich *Servicekomposition* des *Service Oriented Computing* [4] werden spezifische interoperable Technologien eingesetzt, um atomare Funktionen lose gekoppelter Systeme (Webservices) durch Prozesse zu verbinden. Systeme wie eFlow [29] und SELF-SERVE [30] liefern Beispiele. Erweiterte Modelle beziehen Webservice *Konversationen* [31] und komplexere Prozessmodelle [32] ein. FRESCO nutzt ebenfalls die grundlegenden Techniken der Servicekomposition und erweitert diese durch ein konzeptionelles Kooperationsmodell, das mit seinen dienstorientierten Konzepten der expliziten, zusammengesetzten Fähigkeiten mit Trennung von Inhalt und Erbringung über bisherige Ansätze hinausgeht.

## 5 ZUSAMMENFASSUNG UND AUSBLICK

Eine zunehmende Menge von Organisationsformen aus Wirtschaft, Verwaltung, Bildung usw. ist durch intensive inter-organisatorische Beziehungen gekennzeichnet. Unser Fokus liegt auf der komplexen Vielzahl zusammenhängender Interaktionsprozesse bei dienstorientierten *Kooperationsmustern*. Während die technische Basis zur Kopplung der Teilnehmer hier bereits vorhanden ist, fehlt es bislang an Modellen und Mechanismen zur effektiven Planung, Konstruktion und Betrieb entsprechender Kooperationsanwendungen.

FRESCO bietet ein umfassendes Rahmenwerk zur Unterstützung dienstorientierter Kooperationsanwendungen, das ein konzeptionelles Dienstmodell, ein konkretes Architekturmodell und eine DMS Plattform beinhaltet. Der konzeptionelle Teil des Rahmenwerks schlägt die Strukturierung vielschichtiger Interaktionsprozesse in geordnete Kooperationsmuster vor, die zum einen die Modellierung komplexer Beziehungen erleichtern und sich zum anderen effizient durch Workflowkonzepte umsetzen lassen. Das praktische Managementkonzept umfasst deren Spezifikation, Adaption, Aggregation und Koordination.

Der technische Teil des Rahmenwerks setzt Kooperationsmuster und DMS in einer erweiterten Grid Umgebung um und integriert dazu effektive webservicebasierte Prozessstechniken in die Open Grid Service Architecture. Diese Konzepte wurden in einem prototypischen Rahmenwerk implementiert, wobei sich ihre Realisierbarkeit und Stimmigkeit bestätigt hat. In weiterführender Arbeit sollen nun erweiterte Modelle und Mechanismen zur Komposition von Verbunddienstleistungen untersucht werden, die es erlauben, die Dienstfähigkeiten von Verbunddiensten mit den zugrunde liegenden Dienstfähigkeiten der Dienstkomponenten in Beziehung zu setzen und diese zu verbinden.

## LITERATUR

- [1] B. Medjahed, B. Benatallah, A. Bouguettaya, A. H. H. Ngu, and A. K. Elmagarmid, "Business-to-business interactions: issues and enabling technologies," *The VLDB Journal*, (Springer, Online First, April 3, 2003), 2003.
- [2] ISO/IEC-JTC1/SC21, "Basic Reference Model of Open Distributed Processing, Parts 1-4," International Organisation for Standardization 10746, 1995.
- [3] G. D. Michelis, E. Dubois, M. Jarke, F. Matthes, J. Mylopoulos, M. P. Papazoglou, K. Pohl, J. Schmidt, C. Woo, and E. Yu, "Cooperative Information Systems: A Manifesto," in *Cooperative Information System: Trends and Directions*, G. Schlageter, Ed.: Academic Press, 1997.
- [4] M. P. Papazoglou and D. Georgakopoulos, "Service-oriented computing: Introduction," *Communications of the ACM*, vol. 46, pp. 24-28, 2003.
- [5] W3C, "Web Service Activity," <http://www.w3.org/2002/ws/>, 20.2.2003
- [6] A. Tsalgatidou and T. Pilioura, "An overview of standards and related technology in Web Services," *Distributed and Parallel Databases*, vol. 12, pp. 135-62, 2002.
- [7] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *International J. Supercomputer Applications*, vol. 15, 2001.
- [8] F. Curbera, Y. Golan, J. Klein, F. Leymann, D. Roller, S. Thatte, and S. Weerawarana, "Business Process Execution Language for Web Services, Version 1.0," BEA, IBM, Microsoft 31 July 2002.
- [9] G. Piccinelli, C. Zirpins, and W. Lamersdorf, "The FRESCO Framework: An Overview," in *2003 Symposium on Applications and the Internet Workshops (SAINT 2003 Workshops)*: IEEE Computer Society, 2003, pp. 120-123.
- [10] G. Piccinelli, C. Zirpins, and C. Gryce, "An Architectural Model for Electronic Services," University College London, University of Hamburg 2003.
- [11] C. Bussler, "Behavior abstraction in semantic B2B integration," in *Conceptual Modeling for New Information Systems Technologies. ER 2001 Workshops. HUMACS, DASWIS, ECOMO, and DAMA. Revised Papers Lecture Notes in Computer Science Vol. 2465*. 2002, I. Hunt, Ed.: Springer Verlag, Berlin, Germany, 2002, pp. 377-89.
- [12] WfMC, "Workflow Process Definition Interface -- XML Process Definition Language 1.0 Final Draft," Workflow Management Coalition WfMC-TC-1025, October 25 2002.
- [13] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Open Grid Service Infrastructure WG, Global Grid Forum 2002.
- [14] B. Kiepuszewski, A. H. M. ter Hofstede, and W. M. P. van der Aalst, "Fundamentals of control flow in workflows," *Acta Informatica*, vol. 39, pp. 143-209, 2003.
- [15] W. M. P. van der Aalst, "Don't go with the flow: Web services composition standards exposed," *IEEE Intelligent Systems*, vol. 18, 2003.
- [16] C. Bussler, "The role of B2B protocols in inter-enterprise process execution," in *Technologies for E Services. Second International Workshop, TES 2001. Proceedings., Lecture Notes in Computer Science Vol. 2193*, M. C. Shan, Ed.: Springer Verlag, Berlin, Germany, 2001, pp. 16-29.
- [17] D. Georgakopoulos, H. Schuster, A. Cichocki, and D. Baker, "Managing process and service fusion in virtual enterprises," *Information Systems*, vol. 24, pp. 429-56, 1999.
- [18] O. Perrin, F. Wynen, J. Bitcheva, and C. Godart, "A Model to Support Collaborative Work in Virtual Enterprises," in *Business Process Management International Conference, BPM 2003, Eindhoven, The Netherlands, June 26-27, 2003. Proceedings, LNCS 2678*, M. Weske, Ed.: Springer, 2003, pp. p. 104 ff.
- [19] H. Davulcu, M. Kifer, L. R. Pokorny, C. R. Ramakrishnan, I. V. Ramakrishnan, and S. Dawson, "Modeling and Analysis of Interactions in Virtual Enterprises," in *Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises (RIDE'99)*. Washington - Brussels - Tokyo: IEEE, 1999, pp. 12-18.
- [20] J. O. Aagedal, Z. Milosevic, and A. Wood, "Modelling Virtual Enterprises and the Character of their Interactions," in *Proc. Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises (RIDE'99)*. Washington, Brussels, Tokyo: IEEE Computer Society, 1999, pp. 19-26.
- [21] H. Schuster, D. Georgakopoulos, A. Cichocki, and D. Baker, "Modeling and Composing Service-Based and Reference Process-Based Multi-enterprise Processes," in *Proc CAiSE 2000, LNCS 1789*, L. Bergman, Ed.: Springer, 2000, pp. 247-263.
- [22] K. Baina, S. Tata, and K. Benali, "A Model for Process Service Interaction," in *Business Process Management International Conference, BPM 2003, Eindhoven, The Netherlands, June 26-27, 2003. Proceedings, LNCS 2678*, M. Weske, Ed.: Springer, 2003, pp. 261 ff.
- [23] G. Piccinelli, A. Finkelstein, and S. L. Williams, "Service-oriented work-flows: the DySCo framework," in *Proc. 29th EuroMicro Conference, Antalya, Turkey, 2003*.
- [24] D. Georgakopoulos, M. Hornick, and A. Sheth, "An overview of workflow management: from process modeling to workflow automation infrastructure," *Distributed and Parallel Databases*, vol. 3, pp. 119-53, 1995.
- [25] W. M. P. van der Aalst, "Process-oriented architectures for electronic commerce and interorganizational workflow," *Information Systems*, vol. 24, pp. 639-71, 1999.
- [26] Q. Chen and M. Hsu, "Inter-Enterprise Collaborative Business Process Management," Software Technology Laboratory, HP Laboratories Palo Alto HPL-2000-107, 2000.
- [27] M. Mecella, B. Pernici, M. Rossi, and A. Testi, "A Repository of Workflow Components for Cooperative e-Applications," in *Proceedings of the 1st IFIP TC8 Working Conference on E-Commerce/E-Business (Salzburg, Austria, 2001)*: BICE Press, 2001, pp. 73-92.
- [28] E. Colombo, C. Francalanci, and B. Pernici, "Modeling Coordination and Control in Cross-Organizational Workflows," in *Proc. CoopIS/DOA/ODBASE 2002, LNCS 2519*, Z. Tari, Ed.: Springer, 2002, pp. 91 ff.
- [29] F. Casati, M. Sayal, and M.-C. Shan, "Developing E-Services for Composing E-Services," in *Advanced Information Systems Engineering, 13th International Conference, CAiSE 2001, Interlaken, Switzerland, June 4-8, 2001, Proceedings. LNCS 2068*, M. C. Norrie, Ed.: Springer, 2001.
- [30] M. Dumas, B. Benatallah, and Z. Maamar, "Definition and Execution of Composite Web Services: The SELF-SERV Project," *Data Engineering Bulletin*, vol. 25, 2002.
- [31] B. Benatallah, Q. Z. Sheng, and M. Dumas, "The SELF-SERV Environment for Web Services Composition," *IEEE Internet Computing*, vol. January/February 2003, 2003.
- [32] S. Bhiri, O. Perrin, W. Gaaloul, and C. Godart, "An Object Oriented Metamodel for Inter-enterprises Cooperative Processes based on Web Services," in *Proc. Modeling and Developing Process-Centric Virtual Enterprises with Web-Services (VIEWS'03), December, 3-6 Austin Texas, 2003*.