View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Online Publikationen der Universität Stuttgart

Universität Stuttgart

Fakultät Informatik, Elektrotechnik und Informationstechnik

Abschlussbericht des Verbundprojekts Tools4BPEL

Rolf Iron, Oliver Kopp, Andreas Leiner,
Frank Leymann, Peter Massuthe

Report 2009/01
October 08, 2009

Institut für Architektur von Anwendungssystemen

Universitätsstr. 38
70569 Stuttgart
Germany

CR: C.2.4, D.2.2, D.2.4, F.3.2, H.4.1

TOOLS 4 BPEL

Korrektheit und Zuverlässigkeit zusammengesetzter Web Services am Beispiel der Geschäftsprozess-Modellierungssprache BPEL

Abschlussbericht des Verbundprojekts

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter den Förderkennzeichen 01 lSE 08A, 01 lSE 08B, 01 lSE 08C, 01 ISE 08D gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Humboldt-Universität zu Berlin
Institut für Informatik

Universität Stuttgart
Institut für Architektur von Anwendungssystemen

MEGA International GmbH

Kurzfassung

Unternehmensübergreifende Geschäftsprozesse werden zunehmend nach dem Paradigma der Services organisiert. Dabei stellen sich Fragen nach der Komponierbarkeit, Fehlerbehandlung, sowie der Rücksetzbarkeit (Kompensation) im Fehlerfall. In diesem Vorhaben werden Methoden und Werkzeuge zum Umgang mit solchen Fragen entwickelt und am Beispiel der Geschäftsprozess-Modellierungssprache BPEL und im Modellierungswerkzeug der Firma MEGA international erprobt.

Es wurde zum einen der Übersetzer BPEL2oWFN entwickelt, der anhand einer Petrinetzsemantik für BPEL einen BPEL-Prozess in ein (offenes) Petrinetz transformiert. Zum anderen wurden Korrektheitskriterien (wie Bedienbarkeit und Verhaltenskompatibilität) für Services erarbeitet, Algorithmen zu ihrer Überprüfung entworfen und in Fiona implementiert. Die Algorithmen sind Petrinetz-basiert. Damit spielen Übersetzung und Analyse eng zusammen und ein vorhandener BPEL-Prozess kann auf bspw. Bedienbarkeit hin untersucht werden.

In diesem Vorhaben wurden die Modellierungssprache BPEL4Chor, Choreographie-Erweiterungen für BPMN entwickelt, sowie die Transformation von BPMN nach BPEL angepasst, um den Weg von BPMN nach BPEL4Chor zu unterstützen. Weiterhin wurden Konzepte entwickelt, wie sich partner-übergreifende Fehlerbehandlung, Rücksetzbarkeit, sowie die Autonomie der Partner mittels BPEL4Chor darstellen lassen. BPEL4Chor kann als Standardsprache zur Spezifikation von Protokollen, die zwischen mehreren Partnern ablaufen, verwendet werden. Durch seine enge Verbindung mit BPEL kann BPEL4Chor sehr gut als Startpunkt für eine Webservice-Lösung verwendet werden.

Abstract

Cross-company business processes are increasingly organized on the paradigm of services. Specifically, open research questions are about the composability, operability, interchangeability and resettability (compensation) in the case of errors. In this project, methods and tools for dealing with such questions are developed on the basis of the business process modelling language BPEL and tested in the design tool of MEGA International GmbH.

We developed a compiler called BPEL2oWFN to translate a BPEL process into an open Petri net. The compiler builds upon a feature-complete Petri net semantics for BPEL. Furthermore, different correctness notions for services (e.g. controllability of a service and behavioural compatibility of service) were developed, which can be checked on the Petri net level by the tool Fiona. Together with the compiler, we can immediately apply all techniques to BPEL processes.

In this project, the language BPEL4Chor, choreography extensions for BPMN as well as the transformation from these extension to BPEL4Chor were developed. Additionally, capabilities to model cross-process fault handling, cross-process transaction handling as well as autonomy grades were added to BPEL4Chor. BPEL4Chor can be used as standard language for specification of protocols specifying the business protocol running between multiple partners. The tight integration of BPEL allows BPEL4Chor being a good starting point for a Web Service solution.
.

Inhaltsverzeichnis

1. Kurzdarstellung 4

1.1 Stand der Wissenschaft und Technik zu Projektbeginn 6

1.2 Planung und Ablauf des Vorhabens 8

1.3 Zusammenarbeit mit anderen Stellen 11

2. Eingehende Darstellung 14

2.1 Schwerpunkt A: Eigenschaften von Services 14

2.2 Schwerpunkt B: Komposition von Services 16

2.3 Schwerpunkt C: Korrektheit 20

2.4 Schwerpunkt D: Produktreifung und –integration 21

2.5 Entwickelte Werkzeuge 22

2.6 Verwendung der Zuwendung 24

2.7 Notwendigkeit und Angemessenheit der geleisteten Arbeit 24

2.8 Voraussichtlicher Nutzen 25

2.9 Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen 28

2.10 Erfolgte und geplante Veröffentlichungen 29

2.11 Ermöglichte studentische Arbeiten 30

A. Erfolgte und geplante Veröffentlichungen 38

A.1 Dissertationen 38

A.2 Zeitschriftenbeiträge 38

A.3 Konferenzbeiträge 39

A.4 Workshopbeiträge 40

A.5 Technische Berichte 42

A.6 Geplant 43

1. Kurzdarstellung

Die klassische Realisierung der Abläufe in Unternehmen als monolithischer Geschäftsprozess wird in den letzten Jahren in zunehmendem Maße durch verteilte Geschäftsprozesse ersetzt. Diese unterstützen viel besser die flexible Bildung und Umformung loser Kooperationen sowie damit verbundene, neue Prinzipien wie Outsourcing und virtuelle Unternehmen. Technisch gewinnt hier das Konzept der Services immer mehr an Bedeutung. Ein Service kann als eine Menge von Aktivitäten, die in einer Kontrollstruktur angeordnet sind, gesehen werden. Ein Service enthält zudem eine Schnittstelle und einen eindeutigen Bezeichner. Für die Beschreibung der Kontrollstruktur eines Service wird die Sprache „Business Process Execution Language for Web Services“ (BPEL4WS, WS-BPEL, kurz BPEL) [CGK+02, Alv07] verwendet. BPEL wird bereits von vielen bedeutenden Firmen weltweit unterstützt. BPEL selbst ist Teil der Web-Service-Plattform-Architektur [WCL+05], was wiederrum eine technische Umsetzung der Service-orientierten Architektur (SOA) [Got00] ist.

Angetrieben durch die fortschreitende Globalisierung, die Idee virtueller Unternehmen und dank leistungsfähiger Kommunikationsmöglichkeiten über das Internet gewinnen unternehmens*übergreifende* Geschäftsprozesse (im weiteren Verlauf als *partnerübergreifend* bezeichnet) aber zunehmend an Bedeutung. Solche Prozesse können in der Regel nicht als herkömmliche Workflows abgebildet werden, da dies z. B. die unerwünschte Preisgabe von Firmengeheimnissen beteiligter Partner zur Folge hätte. Partnerübergreifende Geschäftsprozesse werden aus Services zusammengesetzt und bilden ihrerseits einen partnerübergreifenden Service. Dies geschieht durch eine (zwischen den beteiligten Partnern auszuhandelnde) *Service-Choreographie* [KBR04,Pel03]. In diesem Projekt wurde die Sprache BPEL4Chor entwickelt. Sie dient der Spezifikation einer solchen Service-Choreographie auf Basis von BPEL. Dabei wurden insbesondere die Aspekte der Beschreibung der Kompositionsstruktur, der Beschreibung von Autonomie, der Beschreibung transaktionaler Zusammenhänge und die Beschreibung einer Fehlerkoordination untersucht.

Die große Herausforderung der Service-Idee selbst ist, dass die später an einem verteilten Geschäftsprozess teilnehmenden Services zunächst unabhängig voneinander entwickelt werden und dann erst dynamisch zur Laufzeit aneinander gebunden werden. Dabei entstehen unter anderem Fragen nach der Kompatibilität zweier Services S und R, also insbesondere ob beide ohne Verklemmungen miteinander interagieren und ein gemeinsames Ziel erreichen können. Darauf aufbauend kann man nach der Sinnhaftigkeit eines Service S fragen, also ob S später überhaupt geeignet ist, mit wenigstens einem anderen Service R interagieren zu können. Diese und andere Fragestellungen wurden im Projekt bearbeitet. Dazu wurden formale Korrektheitskriterien entwickelt und Entscheidungs- und Konstruktionsalgorithmen erarbeitet. Als unterliegende Modellierungssprache wurden offene Petrinetze (auch bekannt als open workflow nets) verwendet, da sie sich für die Modellierung und Analyse verteilter Systeme als besonders geeignet erwiesen haben.

Um die Analyse auch auf BPEL-Prozesse anwenden zu können, ist nun nur noch eine Übersetzung von BPEL in Petrinetze, wie sie im Projekt verwirklicht wurde, notwendig. Somit können die Analysealgorithmen einerseits auf ihre Praxistauglichkeit hin untersucht werden, andererseits ist die

Analyse trotzdem weitgehend losgekoppelt von der steten Weiterentwicklung der konkreten Sprache BPEL. Bei einer neuen BPEL-Version muss nur noch der Übersetzer (wahrscheinlich leicht) angepasst werden, und die Analyse kann sofort wieder angewendet werden.

Da sowohl die manuelle Analyse, als auch die händische Übersetzung selbst sehr fehleranfällig sind, ist hier die Unterstützung von Werkzeugen dringend nötig. Diese wurden im Projekt ebenfalls entwickelt.

Standardisierte Notationen der Prozessmodellierung, wie etwa BPMN, BPEL als Implementierungssprache von Geschäftsprozessen durch Webservices, automatisierte Prozesse durch Workflow-Engines – alle diese Aspekte sind Trends und Entwicklungen des Marktes im Bereich Geschäftsprozessmanagement, denen sich MEGA International stellen und auf die, entsprechend den Kundenanforderungen, eine Antwort gefunden werden muss.

Diese Trends sind zunehmend Kriterien der Kaufentscheidung bzgl. eines Modellierungstools und, auch wenn BPEL und ähnliche Konzepte gegenwärtig noch nicht oder nur wenig genutzt werden, möchten Kunden dennoch „Zukunftssicherheit“ einkaufen hinsichtlich der Automatisierung von Prozessen bzw. der Übergabe der fachlichen Modelle an eine Laufzeitumgebung.

Offensichtlich erscheint jedoch die Diskrepanz zwischen den Anforderungen an eine fachliche Modellierung und den Anforderungen an die Modelle bzgl. ihrer Automatisierbarkeit, oder anders gesagt: Ein fachliche Modellierung und ihre Interpretation durch Menschen, die damit arbeiten, erlaubt mehr Freiheitsgrade in der Darstellung als wenn diese Modelle in eine Laufzeitumgebung übergeben werden sollen. Damit stellt sich für MEGA International die Frage nach den Kriterien für die Güte einer Prozessmodellierung, damit diese Modelle tatsächlich ablauffähig sind.

Die Korrektheit und Zuverlässigkeit von BPEL-Prozessen erscheint für MEGA International daher weniger eine Frage der korrekten Übersetzung von Modellierungsobjekten in BPEL-Objekte zu sein, als vielmehr eine Frage des ursprünglichen Prozessdesigns. Die rein logische Ausführbarkeit der fachlichen Modelle ist nicht gleichbedeutend mit der technischen Ausführbarkeit. Die Frage bleibt also, ob nach wie vor verschiedene Modelle notwendig sind, um den gleichen Gegenstand zu beschreiben. Die Erwartungen hinsichtlich des Projekts zielen daher auf die Analysemöglichkeiten der Projektpartner und entsprechender Hinweise auf die Verbesserung des Prozessdesigns und damit der Modellierung.

Für MEGA International ist ferner die Frage interessant, inwiefern BPEL als Austauschformat für unterschiedliche Fragestellungen und als Notations-unabhängiges Format bei der Übergabe an Workflow-Engines bzw. bei der Übergabe von Informationen zu einem Geschäftsprozess an andere Tools benutzt werden kann. Das Interesse der Zusammenarbeit liegt hier klar bei der Expertise der beiden beteiligten Projektpartner im Zusammenhang mit BPEL. Zudem erhofft sich MEGA im Rahmen dieser Expertise das gemeinsame Erkennen zukünftiger Trends, um darüber Wettbewerbsvorteile realisieren zu können.

Die vom Projektpartner Gedilan Technologies GmbH durchgeführten Teile sind in [Fre07] zu finden. Die Fragestellung des Projekts wurde in [KFL06] zusammengefasst.

1.1 Stand der Wissenschaft und Technik zu Projektbeginn

1.1.1 Analyse von Services

Austauschbarkeit und Abstraktion

Arbeiten zur Austauschbarkeit und über systematische Abstraktion gibt es vor allem im Gebiet Prozessalgebra [BPS01]. Die dort untersuchten Konzepte für Austauschbarkeit von Prozessen sind allerdings für Services aus zwei Gründen ungeeignet. Erstens werden hauptsächlich synchron kommunizierende Systeme untersucht, während für Services (speziell Web Services) eher ein asynchrones Kommunikationsmodell angemessen ist. Zweitens wird Austauschbarkeit als die Nichtbeobachtbarkeit des Austauschs in der Umgebung eines Prozesses betrachtet. Für den Austausch eines Service S1 durch eines Service S2 ist dagegen die Nichtbeobachtbarkeit des Austauschs lediglich für bedienende Umgebungen erforderlich, während Umgebungen, die S1 nicht bedienten, S2 gern bedienen dürfen. Für die Austauschbarkeit von Workflows gibt es Arbeiten von van der Aalst. Er betrachtet allerdings keine Interfaces, so dass seine Resultate das hier gestellte Problem nicht lösen.

Konsistenz

Das Konsistenzproblem kann auf die Untersuchung von Deadlocks in komponierten Systemen zurückgeführt werden. Das Model Checking [CGP99] kennt ausgereifte Lösungen für die Suche nach Deadlocks. Diese Lösungen können für spezielle Anwendungsgebiete durch spezielle Kombination und Konfiguration existierender Techniken optimiert werden. Für Services gibt es solche Optimierungen bisher nicht.

Bedienbarkeit

Bedienbarkeit kann als Controller-Synthese-Problem für ein Discrete-Event-System [RW87] aufgefasst werden. Dabei wird die Umgebung eines Service als Controller aufgefasst, der die Eigenschaft Deadlockfreiheit für das System sicherstellt. Die für Services relevante Problemstellung (asynchrone Kommunikation, Controller hat keinen Zugriff auf interne Zustände des Service) wurde bisher nur als Teilmenge einer wesentlich größeren Klasse von Problemen untersucht. Für diese Klasse sind auch lediglich Unmöglichkeitsresultate bekannt. Für Bedienbarkeit von Services gibt es dagegen bereits einige positive Resultate (siehe eigene Vorarbeiten).

Bedienbarkeit kann alternativ als Model-Cheking-Problem der kürzlich vor-geschlagenen Logik ATL [AHK02] aufgefasst werden. Für diese Logik sind allerdings noch keine Werkzeuge verfügbar.

Martens prägte in seiner Dissertation [Mar04] den für Services zentralen Begriff der Bedienbarkeit, der die Sinnhaftigkeit eines Services formalisiert. Allerdings war die Entscheidung der Bedienbarkeit auf solche Services beschränkt, die ein rein azyklisches Verhalten haben. Erste Arbeiten gab es auch zur Konstruktion von Bedienungsanleitungen für Services [MS05a, MS05b], jedoch ebenfalls auf azyklische Services beschränkt.

Bedienungsanleitungen

Für die automatische Generierung von Bedienungsanleitungen sind keine Arbeiten Dritter bekannt.

1.1.2 Semantik von BPEL und Übersetzung

Formale Semantiken für BPEL

Es gab mehrere Vorschläge für die Formalisierung der Semantik für BPEL. Sie basieren auf verschiedenen Formalismen, darunter

- Endliche Zustandsautomaten [FBS04,FFK04]
- Prozessalgebra [Fer04]
- Abstract State Machines [FGV04]
- Petrinetze [VA05]

Abgesehen von den Semantiken auf der Basis von Petrinetzen und Abstract State Machines, sind die Semantiken unvollständig, vor allem in der Behandlung von Ausnahmeverhalten (Fehlerbehandlung, Kompensation). Eine erste prototypische Implementation der Petrinetzsemantik aus [Sta04] im Werkzeug BPEL2PN existierte [Hin05]. Sowohl die Semantik als der Übersetzer an sich waren daraufhin optimiert, möglichst verständliche Muster zu verwenden und diese streng schematisch zum endgültigen Petrinetzmodell des übersetzten BPEL-Prozesses zusammenzufügen. Dadurch konnten zwar viele Inkonsistenzen in der BPEL-Spezifikation aufgezeigt werden; die Übersetzung resultierte jedoch außerdem darin, dass die entstandenen Petrinetze oftmals zu groß waren, um dann automatisch in vernünftiger Zeit analysiert werden zu können.

Flexible Modellbildung

Die Generierung eines Petrinetzes aus einer BPEL-Spezifikation kann als ein Übersetzungsproblem analog zur Generierung von Maschinencode für höhere Programmiersprachen aufgefasst werden. Solche Übersetzer optimieren in der Regel den generierten Code mit dem Ziel, Laufzeit und Speicherplatz zu sparen. Für diese Art von Codeoptimierung gibt es zahlreiche Techniken (z.B. Statische Analyse [CC77,NNH99], Slicing [Che93]). Eine spezielle Codeoptimierung für den Fall, dass der Zielcode ein Petrinetz und das Optimierungsziel die möglichst effiziente Anwendbarkeit von Analysewerkzeugen ist, ist nicht bekannt.

Statische Analyse

Statische Analyse [CC77,NNH99] beschafft aus dem Programmquelltext Information über das Laufzeitverhalten eines Programms. Diese Information wird bei der Zielcodeoptimierung oder direkt zur Analyse von Eigenschaften verwendet. Methoden zur statischen Analyse gibt es für viele Klassen von Programmiersprachen. Ihre Anwendung erfordert

- die Aufstellung von Datenflussgleichungen zur Beschreibung der Semantik und
- die Wahl geeigneter Abstraktionen für die effiziente Repräsentation und Berechnung von Information

Für BPEL war keine dieser Voraussetzungen erfüllt.

1.1.3 Transaktionen

Das Konzept dessen, was wir heute im IT Umfeld *Tranksaktionen* nennen (*Spheres of Control*), geht zurück auf [Dav78]. In [Gra81] werden die Prinzipien für die Realisierungen unterschiedlicher Arten von Transaktionen dargestellt. Die Prinzipien der Implementierung von Transaktionen in Datenbanksystemen geht auf [HR83] zurück (hier wurde auch das bekannte *ACID* Akronym geprägt);

die hier beschrieben Transaktionen werden heutzutage auch *klasse Transaktionen* oder *ACID Transaktionen* genannt. Die Notwendigkeit und Verwendung von Kompensation in Transaktionen wird in [Tra83] hervorgehoben. [GS87] beschreibt erstmalig ein Konzept (*Sagas*) für ein Transaktionsmodell welches ACID Transaktionen und Kompensation verbindet. Die Berücksichtigung von Baumstrukturen in Transaktionen findet seine bekannteste Ausprägung in den *Nested Transactions* [Mos82] für nicht-kompensationsbasierte Transaktionen und in [GDK+91] für kompensationsbasierte Transaktionen (*Nested Sagas*). Die Ausweitung von Transaktionen auf verteilte Systeme wird in [X/O93] standardisiert. [Elm92] ist eine Bestandsaufnahme der bis Anfang der 90er Jahre entwickelten unterschiedlichen Transaktionsmodelle. Der Stand der Technik auf dem Gebiet wird zu dieser Zeit in [GR93] zusammengefasst. [Ley95a] führt das Konzept der Kompensationssphären ein, welches die Grundlage für das Transaktionskonzept in BPEL wurde. Dieses Konzept wurde in [Ley95b] erweitert, um auch ACID-basierte Transaktionsmechanismen in Geschäftsprozessen unterstützen zu können (*atomare Sphären*), die in der BPEL-Variante BPELJ verwendet werden. Die Mechanismen zur Einbindung von Anwendungstransaktionen in Transaktionen eines Workflowsystems und Verallgemeinerungen dessen zu *stratifizierten Transaktionen* werden in [Ley97] beschrieben.

1.1.4 Partnerübergreifende Geschäftsprozesse

In [WWC92] das Konzept der Megaprogrammierung eingeführt, welches heutzutage als automatisch ablaufende Workflows bezeichnet würde. Die Strukturierung von Mengen von Transaktionen in solche Megaprogramme (genauer: in azyklischen Graphen) kombiniert mit Kompensationsaktionen findet sich in [WR90] (*ConTracts*); hier ist somit der Übergang von Transaktionskonzepten auf Workflow-Technologie anzusiedeln. Die Verwendung von Kompensation in allgemeinen Geschäftsprozessen wird in [Ley95a] ausgearbeitet (*Spheres of joint Compensation*). Die Kombination von automatischen Workflows und kompensationsbasierten Transaktionen ist in [CGK+02] als Standard spezifiziert. [AW01] stellt ein Modell für die abstrakte Beschreibung automatischer Geschäftsprozesse auf, die über mehrere Partner verteilt sind. Einen ersten Ansatz zu einem Standard in diesem Bereich ist in [KBR04] zu finden. Untersuchungen zur Fehlerbehandlung und Transaktionen in partnerübergreifenden Geschäftsprozessen waren lediglich in Ansätzen vorhanden.

1.2 Planung und Ablauf des Vorhabens

Das Vorhaben Tools4BPEL gliedert sich in vier Schwerpunkte:

- Schwerpunkt A: Eigenschaften von Services
- Schwerpunkt B: Komposition von Services
- Schwerpunkt C: Korrektheit
- Schwerpunkt D: Produktreifung und –integration

1.2.1 A Eigenschaften von Services (HU Berlin)

A1 Austauschbarkeit

In diesem AP sollte ein Austauschbarkeitsbegriff auf der Basis von Bedienungsanleitungen entwickelt und Entscheidungsalgorithmen implementiert werden. Es war für Feb-Mai 2006 und Okt-Jan 2008 geplant und wurde in dieser Zeit ausgeführt

A2 Konsistenz

Der grundlegende Korrektheitsbegriff für alle anderen Analysefragen ist die Verhaltenskompatibilität zweier interagierender Services, auch Konsistenz genannt. Neben der Kompatibilität der Schnittstellen der interagierenden Services ist dies vor allem die deadlockfreie Kommunikation der Services. In diesem AP sollten geeignete Modelle zur Abbildung aller relevanten BPEL-Aspekte entwickelt werden und bestehende Algorithmen zur Analyse von Deadlocks angepasst werden. Das AP war für Jun-Sep 2007 und Aug-Sep 2008 geplant und wurde in dieser Zeit ausgeführt.

A3 Bedienbarkeit

In diesem AP sollte die Theorie der Bedienbarkeit so erweitert werden, dass auch Services mit zyklischem Verhalten analysiert werden können. Weiterhin sollten neue Aspekte wie die Bedienbarkeit unter Beachtung semantischer Constraints in die Theorie eingegliedert werden. Alle Algorithmen sollten implementiert werden. Das AP war für Feb-Mär 2006, Jul-Dez 2006 und Okt-Nov 2007 geplant und wurde in dieser Zeit ausgeführt.

A4 Generierung abstrakter Sichten

Eine abstrakte Sicht eines Services S ist im Wesentlichen ein spezieller Service PV(S) (von *public view*), der ein zu S äquivalentes Verhalten hat, aber eine einfachere Struktur. Hier sollten Anwendungsszenarien für abstrakte Sichten herausgearbeitet und die Generierung des Services PV(S) für S implementiert werden. Das AP war für Apr 2007-Jan 2008 geplant und wurde in dieser Zeit ausgeführt.

A5 Bedienungsanleitungen

Die Theorie der Bedienungsanleitungen war ebenfalls für den zyklischen Fall zu generalisieren. Die Algorithmen zur Konstruktion von Bedienungsanleitungen, sowie Entscheidungsalgorithmen, ob ein Service durch eine Bedienungsanleitung charakterisiert wird, sollten implementiert werden. Das AP war für Okt 2005-Jan 2006, Jun 2006-Mär 2007 und Feb-Sep 2008 geplant und wurde in dieser Zeit ausgeführt.

1.2.2 B Komposition von Services (U Stuttgart)

Arbeitspaket	Planung	Ausführung
B1 Strukturierung partner-übergreifender Prozesse	04/2006 bis 09/2006	04/2006 bis 09/2006
B2 Prozessautonomie	10/2006 bis 03/2007	10/2006 bis 03/2007
B3 Fehlerbehandlung	04/2007 bis 09/2007	04/2007 bis 09/2007
B4 Geschäftsprozessübergreifende Transaktionen	10/2007 bis 06/2008	10/2007 bis 06/2008

Tabelle 1 Übersicht der Arbeitspakete

Von April 2006 bis Juni 2008 wurde der Schwerpunkt B des Projekts, die „Komposition“ von Services bearbeitet. Hier ergaben sich keine Abweichungen von der Planung.

Im Arbeitspaket „Strukturierung partner-übergreifende Prozesse“ wurde die Darstellung von Unterprozessen, Choreographien und die Einsatzgebiete erarbeitet. Im Arbeitspaket „Prozess-Autonomie“ wurde die möglichen Autonomien von Prozessen untersucht und mittels WS-Policy modelliert. Im Arbeitspaket „Fehlerbehandlung“ wurde die verteilte Fehlerbehandlung bei Prozessen untersucht. Hier wurden Szenarien erarbeitet, sowie Sprachelemente zur Propagation und mögliche Abschlüsse der Fehlerbehandlung untersucht. Im Arbeitspaket „Geschäftsprozessübergreifende

Transaktionen“ wurde untersucht, wie lokale Transaktionen mit externen Transaktionen wechselwirken und wie globale Transaktionen modelliert werden können.

Eine detaillierte Darstellung findet sich in Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** „**Fehler! Verweisquelle konnte nicht gefunden werden.**“.

1.2.3 C Korrektheit (HU Berlin)

C1 Flexible Petrinetz-Bausteine

In diesem AP sollten pro BPEL-Aktivität mehrere Petrinetz-Muster entwickelt werden, die je nach Analysefrage und Kontext die konkrete Instanz der Aktivität im BPEL-Prozess optimal abbilden. Das AP war für Okt-Nov 2005, Apr-Sep 2006, Jan-Feb 2007, Dez 2007-Jan 2008 geplant und Aug-Sep 2008 und wurde in dieser Zeit ausgeführt.

C2 Statische Analyse von BPEL

Techniken der statischen Analyse sollten benutzt werden, die Einsatzbedingungen der konkreten Muster aus AP C1 auszuwerten. Das AP war für Okt 2006-Sep 2007 geplant und Jan-Jul 2008 und wurde in dieser Zeit ausgeführt.

C3 Flexible Modellbildung

Beide Techniken aus AP C1 und C2 sollten in den Übersetzer integriert werden. Das AP war geplant für Dez 2005-Mär 2006, Sep 2007 und Mär-Jul 2008 und wurde in dieser Zeit ausgeführt.

1.2.4 D Produktreifung und –integration (MEGA International, U Stuttgart)

D1 Challenge Examples

Realistische Beispiele wurden modelliert und den Projektpartnern zur Verfügung gestellt. Einsatz des existierenden BPEL-Exporters für die Generierung von BPEL-Code für die „Challenge Examples“. Die Basis bildeten anonymisierte Kundenprozesse, um wirklichkeitsnahe und mit einem entsprechenden Komplexitätsgrad versehene Beispiele zu generieren. Die breite Kundenbasis von MEGA International deckt dabei eine Reihe unterschiedlicher Branchen ab und liefert einen „repräsentativen“ Querschnitt unterschiedlicher Prozesse. Dieses Arbeitspaket wurde bis Dezember 2006 von der Gedilan Technologies GmbH und ab Januar 2007 von der MEGA International GmbH durchgeführt.

Um wirklichkeitsnahe Eingabedaten für die anderen Arbeitspakete zur Verfügung zu stellen, wurden kontinuierlich Geschäftsprozesse realistischer Größe und Struktur modelliert und erweitert. Dabei wurden typische Geschäftsprozesse verschiedener Branchen berücksichtigt. In Zusammenarbeit mit den universitären Partnern wurden außerdem für die modellierten Challenge Examples praxisrelevante Verifikationsprobleme erarbeitet.

Im Laufe des Projekts ist zu erwarten, dass neue Anforderungen für die Challenge Examples identifiziert werden. Diese Aktivität umfasst die Erweiterung und Anpassung der Challenge Examples an die neuen Anforderungen sowie die Identifizierung zusätzlicher Validierungsprobleme. Es handelt sich hierbei um eine kontinuierliche Aktivität, da die Notwendigkeit für Erweiterungen der Challenge Examples von der Identifizierung neuer Anforderungen an die Modelle abhängt.

Die modellierten Challenge-Examples wurden in BPEL bereitgestellt.

D2 Validierung

In diesem Arbeitspaket wurden die Werkzeugprototypen validiert und Erfahrungen mit den Entwicklern diskutiert. Die Ergebnisse der anderen Schwerpunkte wurden ab März 2006 analysiert und an die Gegebenheiten von MEGA International zugeschnitten und angepasst werden. Die in den Schwerpunkt A und C entwickelten Prototypen wurden anhand von BPEL-Exporten der Challenge-Examples getestet. Anhand der Tests wurden Engpässe analysiert und Vorschläge für eine Erweiterung der Funktionalität mit den universitären Partnern diskutiert.

Diese Aktivität wurde im Jahre 2006 von der Firma Gedilan Technologies GmbH ausführt und ab Januar 2007 von der MEGA International GmbH.

D3: Produktreifung und –integration

Die Software MEGA wird auf die Einbindung der Werkzeuge aus den anderen Schwerpunkten vorbereitet. Es wurden Schnittstellenspezifikationen erstellt, Tests durcheführt und die Weiterentwicklung der Version 1 des BPEL-Importers vorangetrieben.

Erhebung der Schnittstellenspezifikationen auf der Basis des existierenden BPEL-Exporters von MEGA International, die den Projektpartnern zur Verfügung gestellt werden. Der im Arbeitspaket D4 entwickelte Prototyp eines BPEL-Importers wurde in MEGA integriert und getestet. Im Anschluss an diese Aktivität wurden die Challenge-Examples in BPEL bereitgestellt.

D4: BPEL Exporter/Importer

Die Werkzeuge zur Analyse bzw. Modellierung, die in den Schwerpunkten A, B und C entwickelt werden, basieren zum Großteil auf dem Standard BPEL und entsprechenden Erweiterungen. Um die im Schwerpunkt D entwickelten Modelle den zu entwickelnden Werkzeugen der Schwerpunkte A, B und C zu Verfügung zu stellen bzw. Resultate aus den Werkzeugen MEGA zugänglich zu machen, wurde in diesem Arbeitspaket eine BPEL Import-Schnittstelle für MEGA prototypisch realisiert. Es handelte sich um ein iteratives Vorgehen, wobei jede Version des Importers in Arbeitspaket D3 in MEGA integriert und getestet wurde.

D4.4 Modellierung in BPMN

Von Juli 2008 bis September 2008 sollte ursprünglich der Importer und Exporter von Nautilus um die neuen Konstrukte von BPEL4Chor erweitert werden. Da während des Projekts die Firma Gedilan Technologies Insolvenz anmelden musste und die MEGA International GmbH den Teil von Gedilan übernommen hat, musste das Arbeitspaket angepasst werden. Da die MEGA Suite BPMN als Modellierungssprache unterstützt, wurde von Juli bis November 2009 geklärt, wie sich die Choreographie-Konstrukte in BPMN und insbesondere dem Produkt von MEGA einbringen lassen. Weiterhin wurden die Konstrukte anhand der Beispiele von MEGA auf ihre Praxistauglichkeit geprüft. Da dies ein höherer Aufwand bedeutet hat, wurde das Arbeitspaket um zwei Monate verlängert. BPMN selbst steht für „Business Process Modelling Notation“ und ist in [OMG09] beschrieben.

1.3 Zusammenarbeit mit anderen Stellen

Die Arbeitsgruppe der Humboldt-Universität zu Berlin hat eine langfristige Kooperation mit der Eindhoven University of Technology (Prof. van der Aalst, Prof. van Hee) zu Fragen der Modellierung und Analyse von Geschäftsprozessen vereinbart, die in dem gemeinsamen Projekt „B.E.S.T. – Berlin, Rostock, Eindhoven Service Technology“ mündet. Es besteht weiterhin eine stabile Partnerschaft mit

dem IBM Forschungs- und Entwicklungslabor in Böblingen (Simon Moser), dem IBM T.J. Watson-Research-Center in Hawthorne, sowie dem Research Lab der IBM in Zürich. Dieser „direkte Draht“ zu einem der wichtigsten Unterstützer der Sprache BPEL und einem Vorreiter in der formalen Untermauerung der Service-Idee kommen dem Projekt sehr zu gute. Umgekehrt werden die Ergebnisse des Projekts (auch weiterhin) die weitere Entwicklung des BPEL-Standards beeinflussen, und die Analyse von Services wird immer mehr als ein wichtiger Faktor der Qualitätssicherung im Bereich der Softwareentwicklung mit Services wahrgenommen. Nicht zuletzt beteiligen sich Mitglieder der Arbeitsgruppe aktiv an der internationalen Initiative „Petri nets and Pi calculus for business process modeling“.

Die wissenschaftlichen Ergebnisse dieses Projektes werden bestehende Kooperationen mit Alistair Barros (SAP Research Brisbane) Prof. Buchmann (Universität Darmstadt, Deutschland), Prof. Bussler (Galway, Irland), Prof. Dustar (Universität Wien, Österreich), Prof. Fensel (Universität Innsbruck, Österreich), Dr. Khalaf (IBM Research, Hawthorne), Prof. Papazoglou (Universität Tilburg, Niederlande), Prof. Tai (Universität Karlsruhe), Prof. Weerawarana (Universität Colombo, Sri Lanka) und Prof. Weske (Hasso-Plattner-Institut) beeinflussen. Mit diesen Wissenschaftlern besteht seit geraumer Zeit eine Zusammenarbeit auf dem Gebiet „Komposition von Diensten“ mit so unterschiedlichen Schwerpunkten wie „Adaptivität“, „Modellierung“, „Semantik von Dienten“ oder „Middleware für Service-orientierte Architekturen“.

Erkenntnisse aus Diskussionen mit diesen Kollegen haben umgekehrt das Projekt beeinflusst. Insbesondere wurde im Rahmen des Projekts mit dem Hasso-Plattner-Institut (HPI) in Potsdam zusammengearbeitet, die Grundlage für BPEL4Chor erarbeitet, sowie gemeinsam veröffentlicht (DKB08, DKLPW08, DKLW09, DKLW07, DKP07, PDKL07). Zusammen mit IBM Research wurde die Aufteilung von Prozessen und Prozessfragmenten näher untersucht und es entstanden die Veröffentlichungen KKL07a, KKL08a, KKL08b. Zusammen mit SAP Research Brisbane und dem HPI wurden die Grundlagen von Choreographien in DKB08 veröffentlicht. Mit dem EU-Projekt „SUPER“ (Semantics Utilised for Process Management within and between Enterprises) wurde einerseits die alternative Syntax für BPEL, BPELscript, erarbeitet (BKvLL09) und andererseits die Einbettung von Choreographien in einen Enterprise-Service-Bus untersucht (KvLN08). Zusammen mit dem EU-Projekt „TripCom“ (Triple Space Communication) wurden die Gemeinsamkeiten und Unterschiede von Block-strukturierten und Graph-basierten Sprachen untersucht (KMWL08, KMWL09). Gemeinsam mit dem EU-Projekt S-Cube wurde der MDD-Ansatz zur Entwicklung von BPEL-Prozessen zur Koordination in KWM+08a veröffentlicht. Mit dem DFG-Projekt „Nexus“ (SFB-627) wurden Ergebnisse im Feld der EPK-Transformation verfeinert (KWL09b), auf Complex-Event-Processing-Systeme angepasst (WMKL09), sowie das Konzept der Partner-übergreifenden Fehlerbehandlung und Transaktion dargestellt (KWL09a).

Bei den Kunden von MEGA International handelt es sich um Partnerunternehmen in Deutschland, Österreich und der Schweiz. Bei diesen Partnern handelt es sich um Unternehmen verschiedener Größenordnungen, die extensives Fachwissen in unterschiedlichsten Branchen besitzen. Hierzu zählen Finanzdienstleister, Versicherungen, Automobilindustrie, Pharmaindustrie, Krankenhäuser, Chemische Industrie, Öffentliche Verwaltung, usw.

Durch das auf diese Weise vorhandene breite Fachwissen und den branchenunabhängigen Ansatz von MEGA International ist es möglich, MEGA erfolgreich in einer sehr weiten Spanne von Projekten einzusetzen. Darüber hinaus ermöglichen die in solchen Projekten gewonnenen Erfahrungen eine kontinuierliche und praxisorientierte Anpassung des Produktes an die Erfordernisse der einzelnen Branchen.

Dem Projekt Tools4BPEL steht aufgrund dieses Ansatzes eine große Menge von Daten aus der Praxis zur Verfügung, welche sich bei Bedarf auch noch fachgebietsspezifisch gliedern lassen

2. Eingehende Darstellung

2.1 Schwerpunkt A: Eigenschaften von Services

Der Schwerpunkt A umfasst fünf Arbeitspakete: A1 Austauschbarkeit, A2 Konsistenz, A3 Bedienbarkeit, A4 Generierung abstrakter Sichten und A5 Bedienungsanleitungen. Wir werden diese im Folgenden in der Reihenfolge Konsistenz, Bedienbarkeit, Bedienungsanleitungen, Austauschbarkeit und Generierung abstrakter sichten darstellen, da sie inhaltlich aufeinander aufbauen.

2.1.1 A2 Konsistenz

Die *Kompatibilität des Verhaltens* zweier Services S und R ist der grundlegendste Korrektheitsbegriff im Rahmen dieses Projekts [RFL+06, RBF+07, RWB+07]. Dabei nennen wir S und R *verhaltenskompatibel*, wenn in ihrer Interaktion keine Verklemmungen (Deadlocks) auftreten (ihre Interaktion *konsistent* ist) [MS05a, MS05b, MRS05]. Ein Deadlock kann dabei aus verschiedenen Gründen auftreten [LMSW06, LMSW08]: Zum einen kann einer der beiden Services einen internen Deadlock haben, also einen Zustand, in dem sowohl keine interne Aktion des entsprechenden Services, als auch keine Kommunikation mehr auftreten kann. Zum anderen kann einer der beiden Services eine Nachricht an den anderen geschickt haben, die dieser nicht (mehr) verarbeiten kann. Weiterhin kann ein Service, bspw. S, eine interne, nicht kommunizierte, Entscheidung treffen, von der aber das weitere erwartete Verhalten von R abhängt. Reagiert R dann anders als von S erwartet, tritt oftmals einer der beiden vorigen Deadlock-Situationen auf.

In diesem AP haben wir *offene Netze* (auch bekannt als open workflow nets oder service nets) als Modellierungssprache für Services herausgearbeitet [MRS05, MW06, LMW06,LMW07a]. Offene Netze sind eine Variante von Petrinetzen mit ausgezeichneten Plätzen für die Schnittstellen und einer ausgezeichneten Menge an Endmarkierungen. Strukturell haben sich low-level Petrinetze als geeignet gezeigt, alle für die Verhaltensanalyse relevanten Aspekte von Services abzubilden – für praktische BPEL-Prozesse haben die entstehenden Modelle eine für die Analyse beherrschbare Größe [LMW06,LMW07a, MW08].

Die im Petrinetz-Model-Checker LoLA [Sch00] integrierten Algorithmen zur Aufdeckung von Deadlocks wurden angepasst und in das neue Analysewerkzeug für Services, Fiona [LMSW06, LMSW08, MW08, Mas09], integriert.

2.1.2 A3 Bedienbarkeit

Die Bedienbarkeit [Mar04] eines gegebenen Services S ist die Frage nach der Existenz eines Services R, so dass beide verhaltenskompatibel sind. Da es entweder keinen, oder unendlich viele solche R gibt [Mas09], ist die Bedienbarkeitsfrage nichttrivial. Sie wird entschieden durch Techniken der Controller-Synthese, indem zunächst eine Überapproximation des möglichen Verhaltens eines beliebigen Services R erzeugt wird und dann iterativ solches Verhalten von R, dass Deadlocks in der Interaktion mit S erzeugt, aus der Überapproximation gestrichen wird [LMSW06, LMSW08]. Ist das Verhalten nach Beendigung des Streichens leer, dann erzeugt jeder Service R einen Deadlock mit S und S ist somit nicht bedienbar. Ist das Verhalten nicht leer, dann konnte ein Zeuge für kompatibles Verhalten gefunden werden. Die Entscheidung der Bedienbarkeit von S erzeugt also im positiven Fall einen konkreten bedienenden Service R.

In diesem AP wurde die Theorie der Bedienbarkeit (hier im Wesentlichen die algorithmische Konstruktion eines Bedieners) auf den Fall generalisiert, in dem der Service S auch zyklisches Verhalten aufweist [LMW07a]. Wir konnten weiterhin zeigen, dass die Bedienbarkeit von S unentscheidbar ist, wenn S unbeschränktes Verhalten hat [MSSW08]. Ebenfalls haben wir untersucht, wie die Konstruktion eines bestimmten Services R gesteuert werden kann, der gegebene semantische Abhängigkeiten der Nachrichten respektiert (das Ausfüllen eines Formulars ist semantisch abhängig vom Empfang des leerer Formulars) [Ker07, Sch07, LMW06, LMW07b].

Alle Algorithmen wurden in Fiona implementiert [MW08, Mas09]. Diverse bewiesene und implementierte Reduktionstechniken [Mül07, LMSW06, LMSW08] dienen dazu, möglichst kleine Zeugen R zu konstruieren und somit die Bedienbarkeit von S möglichst schnell zu entscheiden.

2.1.3 A5 Bedienungsanleitungen

Die grundlegende Eigenschaft von Service-orientierten Architekturen ist das dynamische Suchen, Finden und Verbinden von Services. Dazu müssen immer und immer wieder neue Paare von Services S und R auf ihre Kompatibilität hin untersucht werden, um eine fehlerfreie Interaktion zwischen ihnen zu gewährleisten. Da die Entscheidung der Kompatibilität wie in Arbeitspaket A2 beschrieben im Wesentlichen auf Techniken des Model-Checkings beruhen, sind sie dazu ungeeignet [Mas09]. Die Bedienungsanleitung OG(S) (von operating guideline) eines Services S ist eine operationelle Charakterisierung aller zu S kompatiblen Services R [MS05a, LMW06, LMW07a, Mas09]. Das bedeutet, dass für jeden beliebigen Service R mit Hilfe von OG(S) getestet werden kann, ob R zu S kompatibel ist. Dieser Test ist wesentlich schneller als die Komposition von R und S auf Deadlocks zu prüfen, da in OG(S) bereits viele Teile der Berechnung fest kodiert sind [MW07, Mas09].

In diesem AP haben wir die Theorie ebenfalls auf den azyklischen Fall generalisieren können [LMW06,LMW07a]. Die Konstruktion von Bedienungsanleitungen, sowie der Test, ob ein Service R durch eine Bedienungsanleitung OG(S) charakterisiert wird, wurden in Fiona implementiert [MW08, Mas09].

Die Berechnung der Bedienungsanleitung OG(S) von S dauert deutlich länger als die Entscheidung der Bedienbarkeit von S. Die Hauptursache dafür besteht darin, dass die verschiedenen Reduktionstechniken für die Entscheidung der Bedienbarkeit nicht angewendet werden können, weil sonst OG(S) nicht mehr *alle* kompatiblen Services R für S charakterisieren würde. OG(S) selbst ist auch deutlich größer als der synthetisierte Zeuge R bei der Entscheidung der Bedienbarkeit von S. Deshalb liegt es nahe, symbolische Methoden zur Berechnung und Kodierung von OG(S) zu untersuchen.

In diesem AP haben wir deshalb weiterhin eine Repräsentation von OG(S) als BDDs entwickelt und Algorithmen implementiert, die Bedienungsanleitung als BDD konstruieren [Kas06a, KMW06, KMW07]. Ebenfalls haben wir den Test der Charakterisierung von R durch OG(S) auf BDD-Basis implementiert [Kas06b]. Es hat sich gezeigt, dass bei linearer Vergrößerung von S, die Größe der expliziten Repräsentation von OG(S) exponentiell steigen kann, während die Größe der BDD-Repräsentation derselben Bedienungsanleitung nur linear steigt. Allerdings ist der initiale Overhead der BDD-Speicherung so groß, dass bei den allermeisten getesteten BPEL-Prozessen die explizite Repräsentation effizienter als die BDD-Variante ist.

2.1.4 A1 Austauschbarkeit

In diesem AP wurde ein grundlegender Austauschbarkeitsbegriff für Services formalisiert: zwei Services S und S' sind austauschbar, wenn sie dieselben kompatiblen Services haben [Bre07, Lis07, Lis08]. Somit ist gewährleistet, dass jeder Service R, der mit dem einen Service fehlerfrei interagieren kann, auch mit dem anderen fehlerfrei funktioniert [ALM+07, ALM+08]. Da die Bedienungsanleitungen OG(S) und OG(S') die Mengen der jeweils kompatiblen Services charakterisieren, konnte ein Entscheidungsalgorithmus der Austauschbarkeit auf Basis von Bedienungsanleitungen entwickelt und in Fiona implementiert werden [SMB09].

2.1.5 A4 Generierung abstrakter Sichten

Der public view (auch abstrakte Sicht) PV(S) von S ist ein spezieller Service S', der austauschbar zu S ist. PV(S) soll dabei insbesondere wichtige Interna von S ausblenden, die Geschäftsgeheimnisse oder andere Details von S darstellen. Ein anderes Anwendungsfeld von public views ist beispielsweise die Dokumentation von S [ALM+07, ALM+08].

In diesem AP haben wir einen Konstruktionsalgorithmus für einen public view PV(S) von S erarbeitet, der aus der Bedienungsanleitung von S abgeleitet werden kann [Lau07, ALM+07, ALM+08, SMB09]. Die wesentlichen Vorteile durch dieses Verfahren sind, dass der public view kanonisch ist (jede zwei austauschbaren Services haben denselben public view) und tatsächlich alle Interna versteckt werden (nach der Konstruktion des public views ist nicht mehr erkennbar, aus welchem dieser untereinander austauschbaren Services der public view berechnet wurde) [Mas09].

Die Konstruktion PV(S) ist ebenfalls in Fiona implementiert [MW08, Mas09].

Der Schwerpunkt B umfasst drei Teile: die Integration mit Nautilus, die Komposition von Services, sowie die Modellierung von Choreographien mittels BPMN.

2.2 Schwerpunkt B: Komposition von Services

2.2.1 B1 Strukturierung partner-übergreifender Prozesse

Orchestrierungen betrachten die Zusammenschaltung von Services zu einem Prozess. Im Modell ist das Verhalten der Dienste nicht mit aufgenommen. Ein Dienst selbst kann auch als Prozess realisiert sein. Falls dieser Prozess einen logischen Eingang und einen logischen Ausgang hat und diese von dem Aufrufer verwendet werden, ist dieser Prozess ein *Unterprozess*. Bisher war es nicht vorgesehen, dass der Unterprozess mit dem Aufrufer kommuniziert oder vor/nach dem Aufruf läuft. Dieses Spektrum von Unterprozessen planen wir in [UEKL09] detailliert zu beschreiben.

Bei Choreographien wird das Zusammenspiel mehrerer Prozesse betrachtet. Hier bildet sich ein sogenannter globaler Prozess. In dem Projekt haben wir die Modellierungssprache BPEL4Chor entwickelt, mit denen sich eine Choreographie mittels BPEL beschreiben lässt. Insbesondere besitzt BPEL4Chor keine Abhängigkeiten zu WSDL. Technische Details werden für eine Beschreibung einer Choreographie nicht benötigt. BPEL4Chor als Konzept und Sprache ist in [DKLPW08] und [KL08] beschrieben. BPEL4Chor definiert *Verbindungsmodelle* (interconnection models), bei denen die Aktivitäten von Prozessen durch message links verbunden sind. In [Kip06] wurde untersucht, wie mittels BPEL ein Interaktionsmodell dargestellt werden kann.

Eine Choreographie beschreibt stets die Reihenfolge der ausgetauschten Nachrichten. Zusätzlich kann bei Verbindungsmodellen das interne Verhalten abstrahiert dargestellt werden. Letzteres wird von Interaktionsmodellen unterstützt. Sprachen, die das Modellieren von internem Verhalten nicht zulassen und den Schritt des Nachrichtenaustausches als atomar ansehen, werden *Interaktionsmodelle* (interaction models) genannt. Wir haben in [DKB08] diesen Unterschied herausgearbeitet. In [DKLW09] haben wir Kriterien für Choreographiesprachen herausgearbeitet, sowie gängige Choreographiesprachen evaluiert. In [KL09] haben wir die Frage diskutiert, ob eine Choreographiesprache die Modellierung von internen Aktivitäten erlauben sollte. Wir bejahen dies, da so beispielsweise Entscheidungsgrundlagen der Teilnehmer offengelegt werden können.

Sobald eine Choreographie modelliert ist, möchte man auch zur Ausführung kommen. Hierzu müssen notwendige Details den Prozessen hinzugefügt werden. Zu den Details gehören insbesondere die WSDL-Informationen, damit Dienste eingebunden werden können. Der detaillierte Weg ist in [RKDL08] und [Rei07] beschrieben. Die internen Aktivitäten der Prozesse müssen gegebenenfalls angepasst werden, um ausgeführt zu werden. Dazu kann einerseits die übliche XML-Syntax von BPEL (und BPEL-Editoren) herangezogen werden oder die von uns entwickelte Syntax BPELscript [BKvLL09].

In einer Choreographie können Referenzen übergeben werden. Kennt beispielsweise ein Partner A die Partner B und C, aber Partner B und Partner C kennen sich gegenseitig nicht, so muss A an B die Referenz von C übergeben. Somit kann B an C eine Nachricht senden. Wir haben in [DKP07] gezeigt, wie man diese Referenzübergabemöglichkeit mittels des Pi-Kalküls modellieren kann.

In Bezug auf die Definition von Kontrollfluss unterstützt BPEL sowohl die Definition von Graphen als auch die Definition von Block-orientierten Konstrukten. Da Graph-orientierte und block-orientierte Programmierung grundsätzlich verschieden sind, sich jedoch ein Prozessmodellierer entscheiden sollte, welchem Paradigma er folgen möchte, haben wir in [KMWL08] und [KMWL09] herausgearbeitet, was die beiden Paradigmen unterscheidet und welche Stärken und Schwächen die Paradigmen jeweils besitzen.

Aufbauend auf den Ergebnissen dieses Projektteils wurde in [Ste07] untersucht, wie sich aus bestehenden Orchestrierungen eine Choreographie-Sicht ableiten lässt. Im Gegensatz zu dem Weg von einer Chororaphiebeschreibung (in BPEL4Chor) zu BPEL wird hier von BPEL zu BPEL4Chor gegangen. Mit dieser Arbeit ist ein Anfang im Feld des „Choreography-Reengineering“ getan.

Das Resultat dieses Arbeitspakets beantwortet die Frage nach der Beschreibung der Kompositionsstruktur.

2.2.2 B2: Prozessautonomie

Partnerübergreifende Geschäftsprozesse stellen hohe Anforderungen an die Verfügbarkeit der über die Partner verteilten Services um letztendlich die Integrität und Konsistenz des gesamten Geschäftsprozesses gewährleisten zu können. Dennoch müssen die Teile der beteiligten Partner einen hinreichenden Grad an Autonomie haben, da die Partner im Allgemeinen selbständige Entitäten sind. Wir haben Ereignisse aufgezeigt, die Abhängigkeiten zwischen Prozessen etablieren. Zu diesen Ereignissen zählt beispielsweise der Wunsch, einen Prozess anzuhalten. Dann muss auch der Partner-Prozess angehalten werden. Wir haben die Ereignisse in Anforderungen und Angebote

aufgeteilt und eine Autonomie-Taxonomie erstellt, die die Eigenverantwortlichkeit der Teilnehmer nach der Art des Zugriffs kategorisiert. Diese Kategorisierung hilft, die richtige Autonomie für den richtigen Zweck zu finden und neue Autonomiegrade zu kategorisieren und zu ordnen. Die Taxonomie wird in [UEKL09] gezeigt. Eine Umsetzung in eine Laufzeitumgebung mittels WS-Policy ist in [Bis07] beschrieben. Als Vorarbeit wurde in [Ma06] ein prototypischer Editor für WS-Policies entwickelt. Dieser Editor wurde im Arbeitspaket weiterentwickelt, um eine Modellierung für Autonomie-Assertions zu unterstützen.

Ein interessanter Aspekt von Autonomie ist die Möglichkeit der Einbettung der Choreographie in einen Enterprise-Service-Bus [Ley05]. Ein Partner kann sich möglicherweise bei seiner Implementierung nicht an die vereinbarte Choreographie halten. Nun kann der Bus die Nachricht aufhalten oder dem Prozess zurücksenden. Damit hat der Prozess die Autonomie des Nachrichtensendes aufgegeben bzw. gestattet, dass der Bus ihm fehlerhafte Nachrichten zurücksendet. Eine ausführliche Diskussion findet sich in [KvLN08].

Das Resultat dieses Arbeitspaket beantwortet die Frage nach der Beschreibung der Autonomie.

2.2.3 B3 Fehlerbehandlung

Bei einem Fehler innerhalb eines einzelnen Service P wird zunächst dort versucht, diesen zu beheben. Falls die Fehlerbehebung scheitert, ist eine Propagierung des Fehlers auf eine Teilmenge der Partner von P notwendig. Die daraus entstehende verteilte Fehlerbehandlung erfordert Koordination der Partner um nach dem Abschluss der einzelnen lokalen Fehlerbehandlungen die beteiligten Services global konsistent fortführen zu können. Dadurch wird die Korrektheit und Zuverlässigkeit partnerübergreifender Geschäftsprozesse wesentlich erhöht.

Lokale Fehler können von den unteren Schichten des Web-Service-Stacks herrühren. In [FA08] wurde beschrieben wie Fehler von den unteren Schichten auf die oberste Schicht, die Prozessschicht, propagiert werden können.

Durch das Konstrukt der „choreography sphere“ wird eine Propagation und eine Koordination der Fehler ermöglicht. Da die Fehlerbehandlung einer choreography sphere der Fehlerbehandlung eines BPEL-Prozesses entspricht, ist die Semantik leicht verständlich. Details zu der choreography sphere sind in [KWL09a] zu finden. Die Unterstützung des Konzepts im Web-Service-Stack wurde in [Ruf07] und [Ste08] skizziert. Um eine choreography sphere zu steuern, wird ein Koordinator benötigt. Wir haben in [KWM+08a] und [KWM+08b] gezeigt, wie ein solcher Koordinator aus der graphischen Beschreibung eines Koordinationsprotokolls generiert werden kann. Aufbauend auf diesen Ergebnissen wurde in [Bis09] gezeigt, wie sich diese Konzepte bei Interaktionsmodellen verwenden lassen.

Ein Koordinator muss in eine bestehende Softwarelandschaft einsortiert werden oder es muss ein bestehender Koordinator wiederverwendet werden. Als ersten Schritt in diese Einsortierung zeigen wir in [KELU07] wie man von Geschäftsprozessen zu Softwarelandkarten kommt.

Bei der Fehlerbehandlung können entweder die Standard-BPEL-Fehlerbehandlungsroutinen verwendet oder eigene Fehlerbehandlungsroutinen definieret werden. Diese „Rouine“ ist wieder ein Teil des globalen Prozesses. Da eine choreography sphere mehrere Partner umfasst, ist nicht

automatisch klar, zu welchem Partner die Aktivitäten zur Fehlerbehandlung gehören. Eine Möglichkeit ist, diese Aktivitäten bei einem weiteren Partner (oder dem Koordinator) ablaufen zu lassen. Da dieser Partner im Fehlerfall potentiell auf alle Daten der beteiligten Prozesse Zugriff hat, möchte man dies aus Wettbewerbsgründen vermeiden. Deshalb schlagen wir vor, dass die Aktivitäten wie in [KL06] und [Kha08] beschrieben in Fragmente aufzuteilen. Zur Erhaltung der Datenkonsistenz müssen die benötigten Daten zwischen den Fragmenten ausgetauscht werden. Das genaue Vorgehen haben wir in [KKL07a] und [KKL08a] beschrieben. Um festzustellen, welche Daten überhaupt ausgetauscht werden müssen, muss eine Datenflussanalyse durchgeführt werden. Diese haben wir in [KKL07b] und [KKL08b] beschrieben. Für diese Analyse ist es wichtig zu wissen, ob sich XPath-Ausdrücke wechselseitig ausschließen. Dies wurde in [Sch07] untersucht. In [Bre08] wurde die Datenflussanalyse erweitert, um alle Konstrukte von BPEL zu unterstützen.

Das Ergebnis des Arbeitspakets beantwortet die Frage nach der Beschreibung der Fehlerkoordination.

2.2.4 B4 Prozess-übergreifende Transaktionen

In einer Ausführung der Fehlerbehandlung eines Geschäftsprozesses kann sich ein Fehler als nicht korrigierbar erweisen; auch kann sich eine Aktivität als undurchführbar herausstellen (beispielsweise die Reservierung eines ausgebuchten Fluges). In solchen Fällen wird es nötig, bereits ausgeführte Teile eines Geschäftsprozesses (in dem einfachen Beispiel etwa die Reservierung eines Hotelzimmers) zurückzusetzen (zu kompensieren). Dies kann die Kompensation weiterer Aktivitäten nach sich ziehen, so dass längere Kompensationsketten entstehen, die sich insbesondere über mehrere Geschäftsprozesse und somit im Allgemeinen über Partner hinweg erstrecken. In diesem Teil wurden Methoden ausgearbeitet, um solche Geschäftsprozessübergreifende Transaktionen zu modellieren und zu überwachen.

Zur Modellierung von Partner-übergreifenden Transaktionen wurde das Konstrukt der „choreography sphere“ wiederverwendet. Durch choreography spheres können beliebige Aktivitäten von beliebigen zu einer Transaktion verbunden werden. Die Infrastruktur stellt dann sicher, dass die Sphere vollständig kompensiert werden kann. Details sind in[KWL09a] beschrieben.

Falls existierende Scope-Grenzen wiederverwendet werden sollen, kann auch die Methode von [TKM04] oder [PML07] verwendet werden. Ähnlich zu choreography spheres müssen auch hier BPEL-Maschinen verwendet werden, die die entsprechenden Erweiterungen unterstützen. Sowohl bei [TKM04] als auch bei [PML07] wird ein zentraler BPEL-Prozess betrachtet und die aufgerufenen Services als opak behandelt. Falls ein aufgerufener Service nun seinerseits auch ein BPEL-Prozess ist, sind die betroffenen Aktivitäten Teil von zwei Transaktionen. BPEL verwendet zur Koordination von Scopes das WS-BA-Protokoll [PML07]. Falls nun eine Aktivität Teil von zwei Transaktionen ist, muss WS-BA um eine Art 2PC-Teil erweitert werden, um zu ermögliche, dass nach einer „commit“-Nachricht die Aktivität selbst melden kann, dass sie kompensieren möchte. Dies war bisher nicht möglich. Details sind in [KML09] und [Kop10] beschrieben.

Für eine formale Beschreibung des Modells von BPEL4Chor und der choreography sphere, benötigten wir eine klare Syntax von BPEL. Diese haben wir in [KML08] definiert.

Das Ergebnis dieses Arbeitspakets beantwortet die Frage nach der Beschreibung der transaktionalen Zusammenhänge.

2.3 Schwerpunkt C: Korrektheit

Der Schwerpunkt C umfasst drei Arbeitspakete: C1 Flexible Petrinetz-Bausteine, C2 Statische Analyse für BPEL und C3 Flexible Modellbildung.

2.3.1 C1 Flexible Petrinetz-Bausteine

Die ursprüngliche Petrinetz-Semantik für BPEL ordnete jeder Aktivität von BPEL genau ein Petrinetz-Muster zu [Sta04, Hin05]. Dieses Muster wurde dann für jede solche Aktivität in jedem BPEL-Prozess verwendet. Durch die Beachtung des Kontextes einer Aktivität im konkreten zu übersetzenden Prozess können jedoch oft einfachere Muster verwendet werden, die die Aktivität in diesem Kontext adäquat wiedergeben. Beispielsweise kann ein kompakteres Muster für eine Aktivität verwendet werden wenn bekannt ist, dass diese nie kompensiert werden muss. Auch beeinflusst das spätere Analyseziel, welche Details im Modell wiedergegeben werden müssen, und welche weggelassen werden können.

In diesem AP haben wir zu jeder Aktivität verschiedene Muster entwickelt und die entsprechenden Voraussetzungen für die Verwendung formalisiert [LMSW06, LMSW08]. Dadurch ist eine neue Petrinetz-Semantik entstanden [Loh08a,Loh08b], die zum einen viel besser für die Analyse geeignet ist, die zum anderen auch die neuen Konzepte der Version 2.0 von BPEL berücksichtigt. Diese Semantik kann noch bis heute als einzige vollständige Semantik für BPEL 2.0 angesehen werden [LVOS09].

Die Muster-Bibliothek ist in den neu entstandenen Übersetzer BPEL2oWFN [Loh07] integriert.

2.3.2 C2 Statische Analyse für BPEL

Um zu entscheiden, welches Muster nun wann angewendet werden kann, haben wir in diesem AP Techniken der Statischen Analyse angepasst und in BPEL2oWFN integriert [Gör08, LMSW06, LMSW08]. Insbesondere haben wir Datenflussgleichungen entwickelt und Techniken der Kontrollfluss- und Datenanalyse angepasst. Dazu werden als initialer Schritt bei der Übersetzung ein abstrakter Syntaxbaum sowie ein Kontrollflussgraph des BPEL-Prozesses aufgebaut.

Um die erzeugten Modelle noch weiter zu vereinfachen, haben wir klassische Abstraktionstechniken für Petrinetze auf unsere Analysefragen angepasst und ebenfalls in BPEL2oWFN integriert [Jan08, Pil08, Loh07].

2.3.3 C3 Flexible Modellbildung

In diesem Arbeitspaket haben wir beide Techniken – statische Analyse und Musterbibliothek – kombiniert. Somit kann die Generierung eines Petrinetzmodells für einen gegebenen BPEL-Prozess anhand der Informationen aus dem AP C2 so gesteuert werden, dass die jeweils kleinsten, noch korrekten, Muster der Bibliothek aus AP C1 ausgewählt werden können [LMSW08]. Durch die anschließende strukturelle Reduktion des generierten Modells kann dieses noch weiter verkleinert werden. Somit können teilweise Modelle erzeugt werden, die um den Faktor 10 kleiner sind als ohne Musterauswahl und Reduktion. Dies hat einen entscheidenden Einfluss auf die anschließende Analysierbarkeit des Prozesses.

2.4 Schwerpunkt D: Produktreifung und –integration

2.4.1 Integration mit Nautilus

Am Anfang des Projekts war die Firma Gedilan Technologies GmbH mit ihrem Modellierungswerzeug „Nautilus“ beteiligt. In dem Werkzeug wurde als Modellierungssprache die „Ereignisgesteuerte Prozesskette“ (EPK, [KNS92]) verwendet. EPKs bestehen aus Funktionen, Ereignissen und Konnektoren. Das Modell der erweiterten ereignisgesgeuerten Prozesskette (eEPK, [STA05]) erlaubt es, an Funktionen weitere Elemente zu assoziieren. In Nautilus kann das beispielsweise die durchführende Person oder die gesendeten und empfangenen Daten sein. In dem Export nach BPEL wurden diese Annotationen in entsprechende BPEL- und WSDL-Konstrukte umgesetzt. So wurde beispielweise jeder Akteur in einem port type umgesetzt.

Die Firma Gedilan hat in der EPK-Notation modellierte „challenging examples“ geliefert. In diesen Beispielen stellten die Ereignisse zwischen Funktionen ausschließlich interne Ereignisse dar. Deshalb wurden diese Ereignisse zu BPEL transition conditions umgesetzt. Das Ergebnis ist in [KUL06] dokumentiert. In [WMKL08,WMKL09] sind einen anderen Weg gegangen und haben Ereignisse stets als externe Ereignisse interpretiert. Mithilfe eines CEP-Systems wurde somit die Verbindung von SOA und EDA ermöglicht und die Methode „SOEDA“ geschaffen. In [KWL09b] haben wir eine Erweiterung für EPKs eingeführt, um interne und externe Ereignisse für die Transformation explizit unterscheiden zu können. Interne Ereignisse werden in transition conditions umgesetzt, während externe Ereignisse in Empfangsaktivitäten umgesetzt werden. In der Arbeit wurde weiterhin gezeigt, wie aus einer EPK eine BPEL4Chor-Choreographie erstellt werden kann.

2.4.2 Integration mit der MEGA Suite

Die wesentlichen Konzepte und Mechanismen, die aus dem Vorhaben resultieren, werden als Erweiterungen von bzw. Werkzeuge für BPEL vorgeschlagen. Eine prototypische Schnittstelle zum Export und Import einer Teilmenge von BPEL für das kommerzielle Modellierungswerkzeug MEGA wurde entwickelt. Hierbei wurde die unterstützte Teilmenge der Export/Import Schnittstelle durch die entsprechenden Konstrukte in MEGA implizit bestimmt.

Die MEGA Modeling Suite dient in erster Linie der Erstellung und Visualisierung von Geschäftsprozessmodellen. MEGA International übernimmt die Rolle als Entwicklungspartner. Der Schwerpunkt von MEGA liegt in der Erstellung und Visualisierung von Unternehmensarchitekturen. In das Tool von MEGA ist bereits eine BPEL-Export-Schnittstelle eingebettet, sodass diese nicht im Rahmen des Projektes neu gestaltet werden muss, um den Projekterfolg über alle Teilaspekte sicherzustellen.

Neben der eigentlichen Modellierung umfasst das Gebiet des Geschäftsprozessmanagements weitere Schwerpunkte, wie z. B. die Simulation der erfassten Prozesse oder die Automatisierung der Prozesse in Form von Workflows. Da diese Themengebiete nicht zur Kernkompetenz von MEGA gehören, ist eine Umsetzung der entsprechenden Funktionalitäten nicht als Eigenentwicklung geplant. Stattdessen beabsichtigt MEGA, durch entsprechende Schnittstellen und durch eine modulare und somit erweiterbare Architektur Anbindungs- und Integrationsmöglichkeiten für externe Komponenten bzw. Programme zu schaffen.

Eine der Aufgaben der Firma MEGA innerhalb des Projekts Tools4BPEL bestand darin, die von den Projektpartnern realisierten Prototypen in die Produkt-Entwicklung einfließen zu lassen. Dies geschieht entweder durch eine direkte Integration der neuen Funktionalitäten in die Software oder durch die Bereitstellung von Zusatzkomponenten, die über externe Schnittstellen an MEGA angebunden werden.

Das technische Ziel von MEGA bestand zum einen darin, im Rahmen des Projekts die Architektur von MEGA so zu erweitern, dass die Integration zusätzlicher Komponenten erleichtert wird. Zum anderen wird durch die Realisierung einer BPEL-Schnittstelle eine standardisierte Anbindung an externe Anwendungen, wie z. B. Simulations- oder Workflow-Systeme, geschaffen. Die damit verbundene Erweiterung des Anwendungsfeldes von MEGA bedeutet einen fundamentalen Mehrwert für diese Software.

2.4.3 Modellierung von Choreographien mittels BPMN

Die MEGA Suite unterstützt unter anderem BPMN als Modellierungssprache. unterstützt, wurde von Juli bis November 2009 geklärt, wie sich die Choreographie-Konstrukte in BPMN und insbesondere dem Produkt von MEGA einbringen lassen. Die allgemeine Darstellung der Choreographie-Konstrukte in BPMN und deren Abbildung auf BPEL ist in [Pfi07], [PDKL07], sowie [DKLPW08] beschrieben. Die spezielle Anpassung an die Bedürfnisse von MEGA wurde in persönlichen Gesprächen erörtert.

Ergebnisse für den Schwerpunkt D

Aus Sicht von MEGA sind die Ziele des Projektvorhabens erreicht worden. MEGA ist jetzt in der Lage, an der Schnittstelle zur Laufzeitumgebungen (Workflow-Systemen, die MEGA selbst nicht anbietet), ein kompletteres Paket anzubieten. Dieses beinhaltet:

- Kriterien und Analysemöglichkeiten für die Güte der Bestimmung von Prozessen
- Steigerung der Effektivität und Effizienz bei der Überleitung von der fachlichen Modellierung zu ablauffähigen Prozessen.

Der entwickelte BPEL-Importer bedeutet einen Technologievorsprung und damit einen Wettbewerbsvorteil gegenüber Anbietern im gleichen Marktsegment. Mit den Tools BPEL Ex-/Importer und den durch die Projektpartner entwickelten Analysetools besteht nun die Möglichkeit, Prozessmodelle ad hoc auf ihre Modellierungsgüte bzgl. der technischen Umsetzung hin zu analysieren und diese ähnlich wie Modellierungsregeln zu behandeln.

2.5 Entwickelte Werkzeuge

Abbildung 1 zeigt die entwickelten Werkzeuge und die Integration mit den Partnern. Als Notation wird FMC [Tab05] verwendet. Türkis hinterlegte Bestandteile wurden im Projekt entwickelt. Oryx und MEGA sind zusätzlich gepunktet, da es nicht von Grund auf entwickelt wurde, sondern die bestehende Modellierungssuite Oryx [DOW08] und MEGA Suite erweitert wurde, um eine Modellierung unserer BPMN-Erweiterungen zu ermöglichen. Es ist nun möglich, in Oryx BPMN zu modellieren und BPEL4Chor zu exportieren. Die BPMN-Erweiterungen wurden MEGA zur Verfügung gestellt. MEGA hat seine Suite dahingehend erweitert, BPEL exportieren und importieren zu können. Die gesamte Werkzeugkette des Projekts und ihre Integration ist in [LK08] beschrieben.

In dem WeSPE-Prototypen wurde gezeigt, wie sich die Autonomiegrade mittels WS-Policy ausdrücken lassen. WS-Policy-Attachment erlaubt nun das annotieren von BPEL-Prozessen mit diesen Policies. Das Ergebnis der Transformation von BPMN nach BPEL4Chor kann mittels BPEL2oWFN in ein Petrinetzmodell konvertiert werden. Dieses Modell bildet den Ausgang für die in [LKLR08] vorgestellte Verifikation.

Im Projekt entstanden die beiden Werkzeuge Fiona [MW08, Mas09] (zur Analyse von Petrinetzmodellen von Services) und BPEL2oWFN [Loh07, Loh08a,Loh08b] (zur Übersetzung eines BPEL-Prozesses in ein entsprechendes Petrinetzmodell). Die Werkzeugkette aus BPEL2oWFN und Fiona erlaubt es, einen gegebenen BPEL-Prozess transparent zu analysieren [LMSW06, LMSW08]. Die Werkzeugkette ist in Abbildung 2 dargestellt.

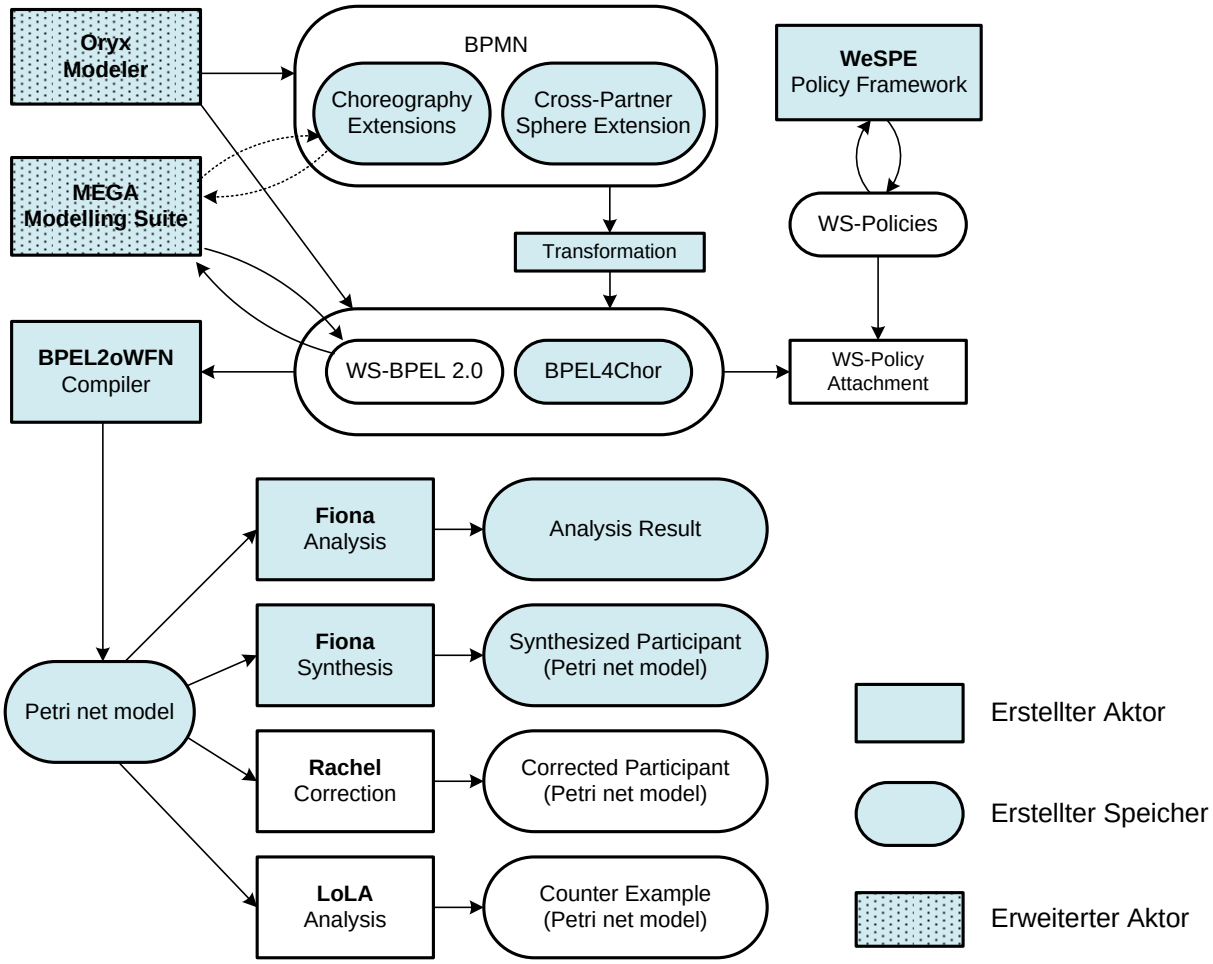

Abbildung 1 Entwickelte Werkzeuge

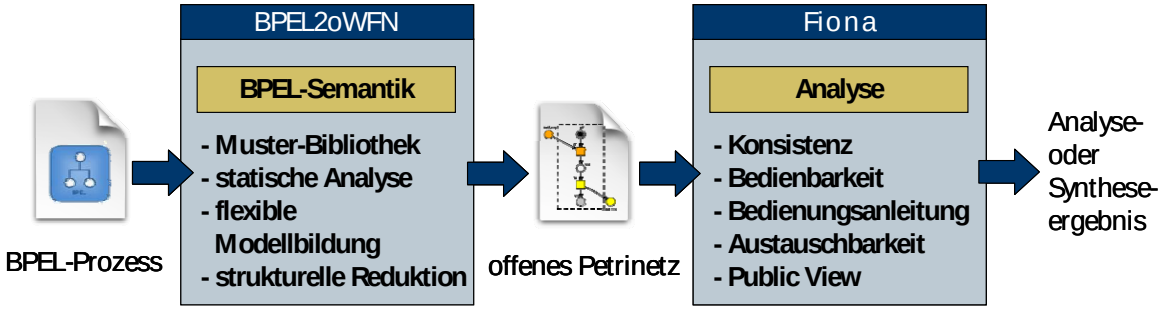

Abbildung 2 Werkzeugkette zur Analyse

Beide Werkzeuge wurden anhand der Challenge Examples insbesondere von MEGA kontinuierlich evaluiert und verbessert. Als Schnittstelle zum Werkzeug von MEGA dient hier der BPEL-Importer und BPEL-Exporter, der ebenfalls im Projekt entstand. BPEL2oWFN und Fiona wurden der MEGA als ausführbare Kommandozeilenprogramme übergeben und können so direkt durch den MEGA Modeler aufgerufen werden.

Um die Aspekte der partnerübergreifenden Transaktionen sowie die Modellierung von Choreographien des Projektpartners IAAS Stuttgart zu unterstützen, wurde der Übersetzer BPEL2oWFN so angepasst, dass auch eine (geschlossene oder offene) BPEL4Chor Choreographie in ein (geschlossenes oder offenes) Petrinetz übersetzt werden kann [LK08, LKLR08]. Da BPEL4Chor sehr stark an BPEL angelehnt ist, musste hierfür die Petrinetz-Semantik für BPEL nur leicht angepasst werden. Somit unterstützt BPEL2oWFN auch die Übersetzung von BPEL4Chor mit den Techniken der flexiblen Modellbildung, was zu sehr kompakten Petrinetzmodellen für BPEL4Chor-Spezifikationen führt.

In den beiden Schwerpunkten A und C wurden alle entwickelten Konzepte in enger Kooperation mit dem industriellen Projektpartner (zunächst Gedilan, später MEGA), aber auch anderen Kooperationspartnern erarbeitet. So konnten wir durch die Formalisierung auf der Ebene von Petrinetzen beispielsweise auch zeigen, dass bestimmte Korrektheitskriterien direkt auf BPEL übertragen werden können, ohne einen Prozess zwischendurch in ein Petrinetz-Modell übersetzen zu müssen [KLM+08].

2.6 Verwendung der Zuwendung

Die Zuwendung wurde verwendet, um die in den vorigen Abschnitten geleistete Arbeit vollbringen zu können. Dafür wurden von der Humboldt Universität zu Berlin 73 PM aufgewendet (davon 37 PM von Mitarbeitern und 36 PM von studentischen Hilfskräften), von der Universität Stuttgart wurden 37 PM aufgewendet, von der Gedilan Technologies GmbH wurden 36 Personalmonate aufgewendet, sowie von der MEGA International GmbH 27 Personalmonate.

2.7 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Durch die Fundierung der Analyse auf das formale Modell der Petrinetze kann die Analyse auch bei sich ändernder BPEL-Spezifikation in der Zukunft leicht wieder verwendet werden. Die Petrinetzsemantik für BPEL ist so modular aufgebaut, dass sie leicht an andere BPEL-Versionen angepasst werden kann (so demonstriert bei der Unterstützung der Version 2.0 von BPEL). Weiterhin können auch (sprachliche) Derivate von BPEL wie BPEL4Chor mit wenig Aufwand unterstützt werden.

Zu Projektbeginn existierte keine Sprache, um Choreographien mittels BPEL auszudrücken. BPEL ist der de facto Standard zur Service-Orchestrierung im Web-Service-Umfeld. Durch die Wiederverwendung von BPEL in der Choreographiemodellierung können existierende Werkzeuge für BPEL wiederverwendet werden, sowie die Erfahrungen, die mit BPEL als Orchestrierungssprache gemacht wurden, auf Choreographiesprachen angewandt werden.

Das Vorhaben war für die Firma MEGA International nicht an einen konkreten Kundenauftrag geknüpft. Obwohl sich die Projektpartner durch die Integration der Werkzeuge in MEGA einen Mehrwert für dieses Produkt versprechen, ist der kommerzielle Erfolg der Methoden mit Risiken behaftet. Um diese Risiken zu minimieren, war die Förderung notwendig und angemessen.

2.8 Voraussichtlicher Nutzen

2.8.1 Wirtschaftliche Erfolgsaussichten nach Projektende

Insbesondere die Fragen der Synthese von kompatiblen Partner-Services und die Entscheidung der Austauschbarkeit von Services werden von MEGA als äußerst wichtiger Wettbewerbsvorteil empfunden. So kann den Partnern eines Geschäftsprozesses vorgeschlagen werden, wie sie sich kompatibel verhalten können, oder welche äquivalenten Alternativen es zu einem Geschäftsprozess es gibt.

Die Modellierung von Choreographien wurde bereits in BPMN umgesetzt und von MEGA adaptiert. Dadurch dass unsere Arbeiten an der Partner-übergreifenden Fehlerbehandlung und Transaktionen beschrieben sind und MEGA durch ihre Partizipation direkt involviert war, kann MEGA die Konzepte rasch in ihr Produkt umsetzen und sich einen Wettbewerbsvorteil verschaffen.

Die entwickelten Konzepte eigenen sich als schnelle Überprüfung der korrekten Modellierung von Services. Dadurch kann die Qualität der modellierten Services wesentlich verbessert werden, bevor die eigentliche Implementierung als realer Geschäftsprozess erfolgt. Dies stellt einen Mehrwert für alle Firmen da, die BPEL unterstützen. Es ist möglich, ein Spin-Off zu gründen, das in diesem Bereich Consulting betreibt.

Durch die formale Zwischenschicht der Petrinetze sind die entwickelten Algorithmen auch nachhaltig anwendbar. Entsteht beispielsweise eine neue Version von BPEL, muss lediglich der Compiler angepasst werden, nicht aber die unterliegende Theorie, oder die Entscheidungsalgorithmen. Sogar ein kompletter Austausch von BPEL durch eine ganz andere Sprache ist somit relativ leicht und schnell zu kompensieren.

Die entwickelten Konzepte eigenen sich als Ideengeber für die Erstellung internationaler Standards. Durch die Konzepte zur Choreographie, insbesondere die Sprache BPEL4Chor, kann sowohl die Sprache BPEL als auch die Sprache WS-CDL oder auch „Let's Dance“ beeinflusst werden. BPEL ist die führende Sprache zur Beschreibung von Geschäftsprozessen in der Service-orientierten Architektur (SOA). Eine Erweiterung von BPEL stellt einen Mehrwert für alle Firmen da, die BPEL unterstützen. Es ist möglich, ein Spin-Off zu gründen, das in diesem Bereich Consulting betreibt.

Da Firmen inzwischen auch nach Choreographie-Lösungen suchen, besitzt die Universität Stuttgart durch die in diesem Projekt entwickelte Sprache „BPEL4Chor“ Know-How aus erster Hand. Mit dem BPMN- und BPEL4Chor-Modellierungswerkzeug besitzt die Universität Stuttgart erste

Werkzeugprototypen. Damit könnte ein Spin-Off für das Consulting in diesem Bereich gegründet und weitere Drittmittelprojekte eingeworben werden.

Durch das Werkzeug „WeSPE“ können nicht-funktionale Anforderungen und Angebote von Prozessen modelliert und verwaltet werden. Das Werkzeug liegt als Prototyp vor. Da neben funktionalen Anforderungen nicht-funktionale Anforderungen bei einer SOA immer wichtiger werden, ist es auch hier möglich, ein Spin-Off zu gründen. Zu den nicht-funktionalen Anforderungen gehören insbesondere die in diesem Projekt entwickelten Autonomiegrade von Prozessen und Services. Die bisherigen Prototypen der Universität lassen keine Arbeit in der Gruppe zu und die Unterstützung von Plugins für neue Typen von Assertions fällt rudimentär aus. Durch „WeSPE“ wäre ein Spin-Off im klaren Wettbewerbsvorteil.

Durch die Integration der BPMN-zu-BPEL-Transformation in Oryx profitiert die Signavio GmbH (http://www.signavio.com) unmittelbar. Sie kann diese Transformation als zusätzlichen Wert verkaufen und sichert somit ihren Eintritt in den Markt und den Standort Brandenburg.

Diese Bestrebungen zur Standardisierung der entwickelten Konzepte werden weitere Fragen aufwerfen, für die die Universität Stuttgart hohe Lösungskompetenzen besitzt. Die Nachfrage nach Werkzeugen zur Unterstützung der Konzepte wird ebenfalls steigen. Auf diese Weise ergeben sich neue wissenschaftliche Fragestellungen, neue Kooperationsmöglichkeiten sowie Chancen zur kommerziellen Verwertung von Folgeprodukten. In Folge dieser Aktivitäten ergeben sich erweiterte Möglichkeiten für die Einwerbung von Drittmitteln aus der Industrie.

2.8.2 Wissenschaftliche und/oder technische Erfolgsaussichten

Die Arbeiten wurden bereits auf international anerkannten Journalen, wie z.B. Data & Knowledge Engineering und „Transactions on Petri Nets and Other Models of Concurrency“, auf Konferenzen, wie z.B. der . der „Web Services and Formal Methods“ (WS-FM), der „Business Process Management“ (BPM) der „International Conference on Applications and Theory of Petri Nets“ (ICATPN), der „International Conference on Advanced Information Systems Engineering“ (CAiSE), der „International Conference on Service Oriented Computing“ (ICSOC), der „International Conference on Services Computing“ (SCC) und der „International Conference on Web Services“ (ICWS) veröffentlicht.

Durch den unmittelbaren Praxisbezug der Ergebnisse sind sie von erheblichem Interesse für die angesprochenen Konferenzen und die hinter ihnen stehenden wissenschaftlichen Forschungseinrichtungen. Es finden bereits auf dem Projektergebnis aufbauende Arbeiten von statt. Beispielsweise wird in [SBFZ08] BPEL4Chor als eine Möglichkeit der Beschreibung von Choreographien angesehen.

Wir können in unseren Projekten auf die Choreographiebeschreibungssprache BPEL4Chor aufbauen. So wird im S-Cube-Projekt BPEL4Chor als SOA-spezifische Choreographie-Sprache verwendet.

Die Resultate werden die zahlreich bestehenden Kooperationen intensivieren und auch neue Kooperationen schaffen. Dies zeigt beispielsweise die Veröffentlichung [Loh08c] der Universität Roststock, wo, basierend auf BPEL4Chor, gezeigt wurde, wie man Fehler in Choreographien halb-automatisch korrigieren kann.

Die gewonnenen Erkenntnisse und die entwickelten Konzepte, insbesondere BPEL4Chor, sind und werden weiterhin Bestandteil von Lehrveranstaltungen der Universität Stuttgart und verbessern somit die Qualität und Aktualität der Lehre. BPEL4Chor und Choreoraphiemodellierung sind beispielsweise fester Bestandteil der Vorlesung „Workflow Management 2“.

2.8.3 Wissenschaftliche und wirtschaftliche Anschlussfähigkeit

Angeregt durch die entstandenen theoretischen Grundlagen im Bereich A1 wurde ein DFG-Projekt speziell zum Thema Austauschbarkeit beantragt und genehmigt, dass das Thema umfassend ausbaut und in dem gerade eine weitere Dissertation entsteht. Weiterhin bilden die Ergebnisse im Bereich der Bedienungsanleitungen (A5) die Basis für ein weiteres DFG-Projekt „Adaptergenerierung“, in dem gerade untersucht wird, unter welchen Umständen zwei Services, die nicht verhaltenskompatibel sind, trotzdem miteinander vernünftig interagieren können.

Bislang lag der Fokus immer auf der Betrachtung der Interaktion genau zweier Services. Dies ist oftmals auch hinreichend für die Analyse der Interaktion einer ganzen Choreographie von n Services, bei der für einen konkreten Services S der Choreographie die verbleibenden (n-1) Services wiederum als ein Partnerservice von S angesehen werden können.

Für viele weitergehende Resultate ist jedoch die *verteilte Bedienbarkeit*, also die Frage nach der Existenz von n Services, die zusammen eine *verteilte* Umgebung für einen Service S darstellen, wichtig. Dies zeigt beispielsweise die Veröffentlichung [LKLR08], wo, basierend auf BPEL4Chor, gezeigt wurde, wie man Fehler in Choreographien halb-automatisch korrigieren kann. Ebenso bei der Frage der Realisierbarkeit einer Choreographie-Spezifikation spielt die verteilte Bedienbarkeit eine zentrale Rolle [LW09]. Hierin sehen wir noch erheblichen Forschungsbedarf.

Die entwickelten Konzepte zur Modellierung von Choreographien werfen Fragen zur Ausführung von Choreographie auf: Wie kann ein Enterpise-Service-Bus dieses unterstützen? Wie kann er helfen, dass die Constraints, die von der Choreographie definiert werden, erfüllt werden? Wie kann er eingreifen, wenn die Constraints verletzt werden? Welche Choreographie-Sprache ist dafür geeignet? Muss eine neue Sprache entwickelt werden? Hierzu sollte eine eingehende Evaluation der bestehenden Arbeiten stattfinden und gegebenenfalls ein eigenes Produkt entwickelt werden.

In diesem Projekt lag der Fokus auf der Modellierung von Prozessen. Der Zusammenhang mit der Architekturlandschaft wurde in einer ersten Publikation in diesem Themenbereich untersucht. Zukünftig gilt es zu untersuchen, welche Zusammenhänge zwischen Choreographien und einer bestehenden Laufzeitumgebung bestehen. Kann hier ein Modellierungswerkzeug für Service-orientierte Architekturen helfen? Hier kann beispielsweise ein Kooperationsprojekt eines Industriepartners und der Universität helfen, diese Fragen zu beantworten.

BPEL setzt sich als Prozessbeschreibungssprache im Bereich der Service-orientierten Architekturen immer mehr durch. Nicht alle BPEL-Prozesse, die zusammenarbeiten, werden als Choreographie beschrieben. Es muss untersucht werden, wie bestehende BPEL-Prozesse durch eine Choreographie beschrieben werden können. Somit ist eine globale Beschreibung der Prozesslandschaft gegeben, die Potential zur Prozessverbesserung und Prozess-Reengineering bietet.

Bisher wurden im Projekt ausschließlich automatisierte Prozesse untersucht. Es stellt sich die Frage, wie „persönliche Prozesse“ [UHW+09] modelliert, analysiert und ausgeführt werden können. Persönliche Prozesse sind Prozesse, die von einem einzelnen Menschen ausgeführt werden und sich adaptiv an die Umgebung anpassen. Dazu sollte sowohl der einzelne Prozess als auch die Choreographie mit der Umgebung untersucht werden. Hier sollte ein zukünftiges, mögliches Projekt untersuchen, welche Möglichkeiten persönliche Prozesse bieten, wie man sie modellieren und ausführen kann. Darauf aufbauend können dann die in dem Projekt entwickelten Verifikationstechniken ansetzen.

Die Firma MEGA International GmbH plant, die im Projekt erarbeiteten Konzepte in ihrem umfassenden Werkzeug zur Prozessmodellierung umzusetzen. Parallel dazu wäre es möglich, dass ein anderer Partner die vom Projekt entwickelten und zur Verfügung gestellten Konzepte und Prototypen in ein weiteres Produkt umsetzt. Eine Ausgründung aus der Universität ist eine vorstellbare Möglichkeit, die Werkzeuge zur Marktreife zu bringen.

2.9 Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

Im Bereich der Übersetzung von BPEL kann die Petrinetzsemantik bis heute als einzige vollständige Semantik für BPEL 2.0 angesehen werden [LVOS09]. Die anderen Semantiken unterstützen entweder nur die veraltete Version 1.1 oder lassen wichtige Aspekte wie Fehler-, Kompensations-, oder Eventhandling aus.

Im Bereich der Analyse von Services beschränken sich die meisten anderen Forschergruppen auf eine synchrone Kommunikation der Services, was die Fragestellungen deutlich vereinfacht – insbesondere im Bereich der Austauschbarkeit von Services [SMB09]. Im Bereich der Bedienbarkeit, Bedienungsanleitungen und public views sind uns keine neuen Arbeiten von Dritten bekannt, ebenso im Bereich der Verifikation von Deadlocks.

In [MLZ08] werden verschiedene Transformationsstrategien von EPKs nach BPEL vorgestellt. Der Fokus liegt auf der Struktur des Prozesses und nicht auf den verschiedenen Umsetzungsmöglichkeiten von Ereignissen (Prozessdaten [KUL06] vs. externe Events [WMKL09]). Bei der Transformation von BPMN nach BPEL hat [ODBH06] ein Vorgehen gezeigt. Wir haben in [PDKL07] und [DKLPW08] darauf aufgebaut und veranschaulicht, wie BPMN erweitert (bzw. konfiguriert) werden muss, um alle Möglichkeiten von BPEL4Chor zu unterstützen und wie die Transformation aus [ODBH06] angepasst werden muss. In [VVK08] wurde eine Alternative zur Transformation von [ODBH06] vorgestellt. Die Arbeit von [GB08] kann dazu dienen, die Art der in [VVK08] bestimmten Strukturen zu bestimmen. Bei einem neuen Release von Nautilus oder Oryx sollten diese Arbeiten beachtet werden. Eine Taxonomie von Modelltransformationen wurde in [MG06] vorgestellt. [SKI08] greift diese Taxonomie auf und stellt alle verfügbaren EPK-nach-BPEL-Transformationen vor.

Die Queensland University of Technology hat zusammen mit SAP eine Choreographie-Sprache „Let's dance“ vorgestellt [ZBDH06]. Sie stellt eine sehr abstrakte Sicht über die Abhängigkeiten zwischen Partnern dar und hat keine Repräsentation in BPEL. Deshalb kann Sie nicht benutzt werden, um „detaillierte“ Abhängigkeiten zwischen Partnern zu modellieren. Eine weitere Choreographiesprache, die dem Interaktionsmodell folgt, wurde in [DB07] vorgestellt. Ein ausführlicher Vergleich von diesen zu BPEL4Chor findet sich in [DKLW09].

In [DP07] wurde eine Choreographie-Erweiterung für BPMN vorgestellt. Wir haben diese Erweiterungen aufgegriffen, weitere hinzugefügt und diese in [PDKL07] vorgestellt. Eine weitere Vorstellung des Konzepts von Choreographien ist in [Dec08] zu finden. In [ZDH+08] wird beschrieben, wie eine Methode zur Entwicklung von Choreographien aussehen kann. Wir haben in [DKLW09] gezeigt, wie wir diese Methode auf BPEL4Chor anwenden können.

In [NLDL07] wurde BPEL^{light} vorgestellt. BPEL^{light} hebt die enge Verzahnung von BPEL zu WSDL auf. Wir haben dieses durch das Konzept des „Groundings“ in BPEL4Chor ebenfalls umgesetzt. Im Gegensatz zu BPEL beschreibt BPEL^{light} den Prozess eines Partners und nicht die vollständige Choreographie.

In [BDH05] sind die Anforderungen an die Ausdrucksfähigkeiten einer Sprache in Bezug auf die möglichen Formen des Nachrichtenaustauschs beschrieben. Wir haben mittels diesen Anforderungen BPEL4Chor in [DKLW07] und [DKLW09] evaluiert und kamen zu dem Schluss, dass praktisch alle Anforderungen durch BPEL4Chor erfüllt sind.

Im Bereich der Fehlerbehandlung wurden unter anderem [AFG+06], [BCCT05], [KVG07] veröffentlicht. In diesen Veröffentlichungen werden ausschließlich Orchestrierungen betrachtet. Eine Betrachtung von Choreographien findet nicht statt. In [CCKM05] wird ein BPEL-Prozess aufgeteilt und eine dezentrale Fehlerbehandlung anhand der BPEL-Semantik durchgeführt. Die Aufteilung von einem BPEL-Prozess in mehrere entspricht der Aufteilung, wie sie in [Kha08] vorgestellt wurde. Im Fall von BPEL4Chor wird die Choreographie direkt und nicht durch eine Aufteilung einer Orchestrierung modelliert. In einer Aufteilung können Scopes sich (per Konstruktion) nicht überlappen, was bei BPEL4Chor möglich ist und in [Ste08], [KWL09a] und [Kop10] beschrieben ist.

Im Bereich der Transaktionen im Web-Service-Bereich bietet der Artikel [WVGK08] eine aktuelle Übersicht. Es gibt bisher keine Veröffentlichungen im Bereich von Transaktionen bei Choreographien. Es wurden, wie z.B. in [SDN08], Erweiterungen von Koordinationsprotokollen vorgeschlagen, jedoch findet keine explizite Modellierung mittels Choreographien statt. In [SA07] wurden Anforderungen an Transaktionsprotokolle gestellt. Als Szenario wurde eine Choreographie gezeigt. Wir haben das Beispiel in [KL09] aufgegriffen und gezeigt, wie man es mittels BPEL4Chor modellieren kann.

2.10 Erfolgte und geplante Veröffentlichungen

Im Rahmen des Projekts ist eine Vielzahl von Publikationen entstanden, die im Anhang A aufgeführt sind.

Zum Schwerpunkt A (Analyse) sind hier insbesondere relevant: [ALM+07, ALM+08, KMW07, KMW07, LMW07a, LMW07b, Mas09, MRS05, MS05a, MS05b, MSSW08, MW06, MW07, MW08, RBF+07, RFL+06, RWB+07, SMB09].

Zum Schwerpunkt B (Komposition von Services) sind hier insbesondere folgende Publikationen relevant: [DKB08, DKLPW08, DKLW09, KKL08a, KL08, KML09, KMWL09, KWL09a, LK08, LKLR08].

Zum Schwerpunkt C (Übersetzung) sind folgende Publikationen zu zählen: [Loh07, Loh08a,Loh08b, LVO+07, LVOS09].

Schwerpunktübergreifende Publikationen sind: [KLM+08, LK08, LKLR08, LMSW06, LMSW08].

2.11 Ermöglichte studentische Arbeiten

2.11.1 Diplomarbeiten

[Bis07] Bischoff, Georg: **Autonomie von Geschäftsprozessen: Beschreibung & Umsetzung**, Diplomarbeit Nr. 2536, 2007, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

[Bis09] Bischof, Marc: **Modeling and Runtime Support of Faults in Interaction Choreography Models**, Diplomarbeit Nr. 2885, 2009, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

[Bre07] Jan Bretschneider. **Produktbedienungsanleitungen zur Charakterisierung austauschbarer Services**. Diplomarbeit, Humboldt-Universität zu Berlin, March 2007.

[Bre08] Breier, Sebastian: **Extended Data-flow Analysis on BPEL Processes**, Diplomarbeit Nr. 2726, 2008, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

[FA08] Fuentetaja Abad, Pablo: **Fault Handling Across the Web Services Stack**, Diplomarbeit Nr. 2728, 2008, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

[Gie08] Christian Gierds. **Strukturelle Reduktion von Bedienungsanleitungen**. Diplomarbeit, Humboldt-Universität zu Berlin, January 2008.

[Gör08] Katharina Görlach. **Ein Verfahren zur abstrakten Interpretation von XPath-Ausdrücken in WS-BPEL-Prozessen**. Diplomarbeit, Humboldt-Universität zu Berlin, March 2008.

[Kas06b] Kathrin Kaschner. **BDD-basiertes Matching von Services**. Diplomarbeit, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, March 2006.

[Ker07] Andreas Kerlin. **Bedienbarkeit unter Kausalität**. Diplomarbeit, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, January 2007.

[Kle07] Jens Kleine. **Transformation von offenen Workflow-Netzen zu abstrakten WS-BPEL-Prozessen**. Diplomarbeit, Humboldt-Universität zu Berlin, July 2007.

[Kip06] Kipp, Alexander: **Ablösung von WS-CDL durch BPEL und WSFL Global Model**, Diplomarbeit Nr. 2379, 2006, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

[Köh08] Patrick Köhnen. **Synthese offener Workflownetze aus Serviceautomaten**. Diplomarbeit, Humboldt-Universität zu Berlin, January 2008.

[Lau07] Peter Laufer. **Public-View-Generierung**. Diplomarbeit, Humboldt-Universität zu Berlin, November 2007.

[Lis08] Nanette Liske. **Laufzeitersetzung offener Workflownetze**. Diplomarbeit, Humboldt-Universität zu Berlin, July 2008.

[Ma06] Ma, Zhilei: **WS-Policy Editor - Ein Werkzeug zum Editieren, Normalisieren, Verschmelzen und Schneiden von Web-Services-Policies**, Diplomarbeit Nr. 2405, 2006, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

[Pil08] Thomas Pillat. **Gegenüberstellung struktureller Reduktionstechniken für Petrinetze**. Diplomarbeit, Humboldt-Universität zu Berlin, March 2008.

[Pfi07] Pfitzner, Kerstin: **Choreography Configuration for BPMN**, Diplomarbeit Nr. 2618, 2007, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

[Ruf07] Ruf, Fabian: **Fault handling in BPEL-based choreographies**, Diplomarbeit Nr. 2598, 2007, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

[Sch07] Schurr, Bastian: **Analyse von XPath-Ausdrücken in BPEL-Prozessbeschreibungen**, Diplomarbeit Nr. 2687, 2007, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

[Ste08] Steinmetz, Thomas: **Ein Event-Modell für WS-BPEL 2.0 und dessen Realisierung in Apache ODE**, Diplomarbeit Nr. 2729, 2008, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

2.11.2 Studienarbeiten

[Ber08] Rico Bergmann. **Vergleich von Werkzeugen zur computergestuetzten Verifiktion von Petrinetzmodellen**. Studienarbeit, Humboldt-Universität zu Berlin, July 2008.

[Bis08] Bischof, Marc: **Translating WS-BPEL 2.0 to BPELscript and Vice Versa**, Studienarbeit Nr. 2175, 2008, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

[Gie07] Christian Gierds. **Ein schärferes Kriterium für die Wahl von Endzuständen in Bedienungsanleitungen, Liberalsten Partnern und Public Views**. Studienarbeit, Humboldt-Universität zu Berlin, October 2007.

[Jan08] Daniel Janusz. **Implementierung zweier Algorithmen zur Abstraktion von Petrinetzen**. Studienarbeit, Humboldt-Universität zu Berlin, April 2008.

[Kas06a] Kathrin Kaschner. **Repräsentation von Bedienungsanleitungen durch BDDs**. Studienarbeit, Humboldt-Universität zu Berlin, January 2006.

[Lis07] Nannette Liske. **Laufzeitersetzbarkeit von Services**. Studienarbeit, Humboldt-Universität zu Berlin, April 2007.

[Mül07] Gerrit Müller. **Strukturelle Analyse von offenen Workflow-Netzen hinsichtlich Bedienbarkeit**. Studienarbeit, Humboldt-Universität zu Berlin, January 2007.

[Rei07] Reimann, Peter: **Generating BPEL Processes from a BPEL4Chor Description**, Studienarbeit Nr. 2100, 2007, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

[Sch07] Alexander Schulz. **Zielgerichtete Strategien**. Studienarbeit, Humboldt-Universität zu Berlin, July 2007.

[Ste07] Steinmetz, Thomas: **Generierung einer BPEL4Chor-Beschreibung aus BPEL-Prozessen**, Studienarbeit Nr. 2101, 2007, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

2.11.3 Fachstudien

In einer Fachstudie untersucht eine Gruppe von Studenten verschiedene Möglichkeiten und gibt eine Empfehlung ab [Lud01]. Untersuchungsgebiete können beispielsweise Werkzeuge, Methoden oder Sprachen sein. In [HWS05] wurden Einsatzszenarien von WS-CDL und BPEL untersucht und eine Empfehlung abgegeben, wann welche Sprache zum Einsatz kommen soll.

[HWS05] Hildebrandt, Ingo; Wagner, Marc; Stützner, Carsten: **Evaluierung verschiedener Szenarien zur Anwendung von BPEL und WS-CDL**, Fachstudie Softwaretechnik Nr. 48, 2005, Institut für Architektur von Anwendungssystemen, Universität Stuttgart.

2.11.4 Studienprojekte

Im Nachlauf zu Tools4BPEL wird das im Rahmen des Aspekts „Prozessautonomie“ entwickelte Werkzeug für WS-Policies gegenwärtig im Studienprojekt WeSPE zu einer Suite ausgebaut. Neun Studenten entwickeln im Team eine Suite, die es erlaubt, WS-Policies in der Gruppe zu entwickeln und an Dienste anzuheften. Dabei liegt das Augenmerk insbesondere auf domänenspezifischen Assertions, die Bestandteil von WS-Policies sind. Eine solche Assertion ist beispielsweise die im Bereich der „Prozessautonomie“ entwickelten Assertions.

Literaturverzeichnis

[AFG06] Ardissono, L.; Furnari, R.; Goy, A.; Petrone, G. & Segnan, M. **Fault Tolerant Web Service Orchestration by Means of Diagnosis EWSA**, Springer, 2006, 4344, 2-16

[Alv07] A. Alves et. al. **Web Services Business Process Execution Language (WS-BPEL) Version 2.0**. Committee specification, OASIS, January 2007.

[AW01] W.M.P. van der Aalst and M. Weske. The **P2P Approach to Interorganizational Workflow**, In Proc. of CAiSE 2001, LNCS volume 2068, Springer, Berlin 2001.

[BCCT05] Brambilla, M.; Ceri, S.; Comai, S. & Tziviskou, C. **Exception Handling in Workflow-Driven Web Applications** In Proc. Int. Conference on World Wide Web. ACM, ACM Press, 2005, 170-179

[BDH+08] Bitsaki, Marina; Danylevych, Olha; van den Heuvel, Willem-Jan; Koutras, George; Leymann, Frank; Mancioppi, Michele; Nikolaou, Christos; Papazoglou, Mike: **An Architecture for Managing the Lifecycle of Business Goals for Partners in a Service Network**. In: Towards a Service-Based Internet, First European Conference, ServiceWave 2008.

[BDH05] Barros, A.; Dumas, M.;ter Hofstede, A. H. M.: **Service Interaction Patterns**. In: Proceedings of the 3rd International Conference on Business Process Management, LNCS 2005, S. 302-318.

[CCKM05] Chafle, G.; Chandra, S.; Kankar, P. & Mann, V. **Handling Faults in Decentralized Orchestration of Composite Web Services**. ICSOC, Springer, 2005, 3826, 410-423

[CGK+02] F. Curbera, Y. Goland, J. Klein, F. Leymann, D. Roller, S. Thatte and S. Weerawarana. **Business Process Execution Language for Web Services BPEL4WS**, BEA Systems & IBM Corporation & Microsoft Corporation 2002 http://www-106.ibm.com/developerworks/library/ws-bpel/

[Dav78] C.T. Davies. **Data processing spheres of control**. IBM Systems Journal 17, 179-198, (1978).

[DB07] Gero Decker and Alistair Barros. **Interaction Modeling using BPMN**, Proceedings of the 1st International Workshop on Collaborative Business Processes (CBP 2007)

[Dec08] Gero Decker. **Choreografiemodellierung - Eine Übersicht**. Informatik Spektrum. pp. 161-166, April 2008

[DOW08] Gero Decker, Hagen Overdick and Mathias Weske. **Oryx - An Open Modeling Platform for the BPM Community**. Demo Session of the 6th International Conference on Business Process Management (BPM), LNCS 5240, pp. 382-385, Milan, Italy, September 2008. Springer Verlag.

[DP07] Gero Decker and Frank Puhlmann. **Extending BPMN for Modeling Complex Choreographies**. In CoopIS 2007, LNCS, Vilamoura, Portugal, November 2007. Springer-Verlag.

[Elm92] A.K. Elmagarmid. **Database Transaction Models for Advanced Applications**. San Mateo, CA, Morgan Kaufmann 1992.

[Fre07] C. Frenkler. **Abschlussbericht der Gedilan Technologies GmbH zum Vorhaben: Korrektheit und Zuverlässigkeit zusammengesetzter Web Services am Beispiel der Geschäftsprozess-Modellierungssprache BPEL**

[GB08] L. García-Bañuelos. **Pattern Identification and Classification in the Translation from BPMN to BPEL**. In On the Move to Meaningful Internet Systems: OTM 2008, volume 5331/2008 of Lecture Notes in Computer Science, pp. 436-444. Springer Berlin / Heidelberg, 2008. doi:10.1007/978-3-540-88871-0 30.

[GDK+91] H. Garcia-Molina, D. Gawlick, J. Klein, K. Kleissner, and K. Salem, **Modeling long running activities as nested sagas**, Data Engineering 14(1), 1991.

[Got00] K. Gottschalk. **Web Services Architecture Overview**. IBM whitepaper, IBM developerWorks, 01 September 2000. http://ibm.com/developerWorks/web/library/w-ovr/.

[GR93] J. Gray, A. Reuter. **Transaction processing**, Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[Gra81] J. Gray. The Transaction Concept: **Virtues and Limitations**. In Proc. VLDB'81, 144 – 154, 1981.

[GS87] H. Garcia-Molina, K. Salem. **Sagas**. Proc. SIGMOD: ACM 1987.

[Hin05] Sebastian Hinz. **Implementierung einer Petrinetz-Semantik für BPEL**. Diplomarbeit, Humboldt-Universität zu Berlin, March 2005.

[HR83] T. Härder and A. Reuter. **Principles of transaction-oriented database recovery**. ACM Computing Surveys 15, 287-317 (1983).

[KBR04] N . Kavantzas, D. Burdett, and G. Ritzinger (ed). **Web Services Choreography Language (WS-CDL1.0)**, online at http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427.

[Kha08] Khalaf, Rania: **Supporting business process fragmentation while maintaining operational semantics : a BPEL perspective**, Dissertation, Universität Stuttgart, 2008.

[KKL07] Khalaf, Rania; Karastoyanova, Dimka; Leymann, Frank: **Pluggable Framework for Enabling the Execution of Extended BPEL Behavior**. In: Proceedings of the 3rd International Workshop on Engineering Service-Oriented Application (WESOA 2007), Springer-Verlag, September 2007

[KL06] Rania Khalaf and Frank Leymann. **Role-based Decomposition of Business Processes using BPEL**. In Proceedings of the IEEE International Conference on Web Services (ICWS '06), pages 770–780. IEEE Computer Society, 2006.

[KNS92] Keller, G.; Nüttgens, N.; Scheer, A.-W.: **Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK)**. Technical Report, Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi), Heft 89, Universität des Saarlandes, 1992.

[KVG07] Kareliotis, C.; Vassilakis, C. & Georgiadis, P. **Enhancing BPEL scenarios with Dynamic Relevance-Based Exception Handling ICWS**, IEEE Computer Society, 2007, 751-758

[Ley05] Leymann, Frank: **The (Service) Bus: Services Penetrate Everyday Life**. In: Service-Oriented Computing - ICSOC 2005, Springer-Verlag, 2005.

[Ley95a] F. Leymann. **Supporting business transactions via partial backward recovery in workflow management systems**, In Proc. BTW'95 (Dresden, Germany, March 22-24, 1995), Springer 1995.

[Ley95b] F. Leymann. **Workflows make objects really useful**, In Proc. HPTS'95 6th Intl. Workshop on High Performance Transaction Systems (Pacific Grove, California, September 17-20, 1995); reprint in EMISA Forum 1996(1).

[Ley97] F. Leymann. **Transaction support for workflows** (in German), Informatik Forschung und Entwicklung 12(1) 1997.

[Loh08c] Niels Lohmann. **Correcting Deadlocking Service Choreographies Using a Simulation-Based Graph Edit Distance**. In Marlon Dumas, Manfred Reichert, and Ming-Chien Shan, editors, Business Process Management, 6th International Conference, BPM 2008, Milan, Italy, September 1-4, 2008, Proceedings, volume 5240 of Lecture Notes in Computer Science, pages 132-147, September 2008. Springer-Verlag.

[Lud01] Ludewig, J. (Hrsg.): **Praktische Lehrveranstaltungen im Studiengang Softwaretechnik: Programmierkurs, Software-Praktikum, Studienprojekte, Fachstudie**. mit Beiträgen von Stefan Krauß, Jochen Ludewig, Patricia Mandl-Striegnitz, Ralf Melchisedech, Ralf Reißing. Bericht der Fakultät Informatik, Universität Stuttgart, 2. Auflage, 2001

[LW09] Niels Lohmann, Karsten Wolf. Realizability **is Controllability**. WS-FM 2009, Springer-Verlag.

[Mar04] Axel Martens. **Verteilte Geschäftsprozesse - Modellierung und Verifikation mit Hilfe von Web Services**. Dissertation, Humboldt-Universität zu Berlin, 2003. Erschienen in WiKi: Stuttgart, Berlin & Paris, 2004.

[MG06] T. Mens, P. V. Gorp. **A Taxonomy of Model Transformation**. Electronic Notes in Theoretical Computer Science, 152:125{142, 2006. doi:dx.doi.org/ 10.1016/j.entcs.2005.10.021

[MLZ08] J. Mendling, K. B. Lassen, U. Zdun. **On the Transformation of Control Flow between Block-Oriented and Graph-Oriented ProcessModeling Languages**. Int. J. Business Process Integration and Management (IJBPIM), 3(2):96-108, 2008.

[Mos82] J.E.B. Moss. **Nested transactions and reliable distributed computing**. In Proc. Reliability in Distributed Software and Database Systems, IEEE 1982.

[MRS05] Peter Massuthe, Wolfgang Reisig, and Karsten Schmidt. **An Operating Guideline Approach to the SOA**. *Annals of Mathematics, Computing & Teleinformatics*, 1(3): 35-43, 2005.

[MS05a] Peter Massuthe and Karsten Schmidt. **Operating Guidelines for Services**. In Karsten Schmidt and Christian Stahl, editors, *12. Workshop Algorithmen und Werkzeuge für Petrinetze (AWPN 2005), Proceedings*, pages 78-83, September 2005. Humboldt-Universität zu Berlin.

[MS05b] Peter Massuthe and Karsten Schmidt. **Operating Guidelines - an Automata-Theoretic Foundation for the Service-Oriented Architecture**. In Kai-Yuan Cai, Atsushi Ohnishi, and M.F. Lau, editors, *Proceedings of the Fifth International Conference on Quality Software (QSIC 2005)*, Melbourne, Australia, pages 452-457, September 2005. IEEE Computer Society.

[NLDL07] Nitzsche, Jörg; van Lessen, Tammo; Karastoyanova, Dimka; Leymann, Frank: **BPEL light**. In: Proceedings of the 5th International Conference on Business Process Management (BPM 2007)

[ODBH06] C. Ouyang, M. Dumas, S. Breutel, A. H. M. ter Hofstede. **Translating Standard Process Models to BPE**L. In CAiSE. Springer, 2006.

[OMG09] Object Management Group. **Business Process Modeling Notation (BPMN) Version 1.2**. 2009. http://www.omg.org/spec/BPMN/1.2

[Pel03] Peltz, C. **Web Services Orchestration and Choreography** *IEEE Computer,* 2003, 36, 46-52

[PML07] Pottinger, Stefan; Mietzner, Ralph; Leymann, Frank: **Coordinate BPEL Scopes and Processes by Extending the WS-Business Activity Framework**. In: Meersman, Robert (Hrsg); Tari, Zahir (Hrsg): Proceedings of the 15th International Conference on Cooperative Information Systems (CoopIS 2007)

[SA07] Sun, C. & Aiello, M. **Requirements and Evaluation of Protocols and Tools for Transaction Management** in Service Centric Systems 31st Annual International Computer Software and Applications Conference (COMPSAC 2007), IEEE Computer Society, 2007, 461-466

[Sch00] Karsten Schmidt. **LoLA: A Low Level Analyser**. In Mogens Nielsen and Dan Simpson, editors, *Application and Theory of Petri Nets, 21st International Conference (ICATPN 2000)*, volume 1825 of *Lecture Notes in Computer Science*, pages 465-474, June 2000. Springer-Verlag.

[SDN08] Schäfer, M.; Dolog, P. & Nejdl, W. **An environment for flexible advanced compensations of Web service transactions** ACM Trans. Web, ACM, 2008, 2, 1-36

[SKI08] S. Stein, S. Kühne, K. Ivanov. **Business to IT Transformations Revisited**. In MDE4BPM. 2008

[Sta04] Christian Stahl. **Transformation von BPEL4WS in Petrinetze**. Diplomarbeit, Humboldt-Universität zu Berlin, April 2004.

[STA05] A.-W. Scheer, O. Thomas, O. Adam. Process-Aware Information Systems: Bridging People and Software Through Process Technology, chapter **Process Modeling Using Event-Driven Process Chains**, pp. 119-146. Wiley & Sons, 2005.

[Tab05] P. Tabeling. **Softwaresysteme und ihre Modellierung**. Grundlagen, Methoden und Techniken. Springer, Berlin, 2005.

[TKM04] Tai, S.; Khalaf, R. & Mikalsen, T. A. Jacobsen, H.-A. (ed.) **Composition of Coordinated Web Services** Middleware 2004, ACM/IFIP/USENIX International Middleware Conference, Springer, 2004, 3231, 294-310

[Tra83] I.L. Traiger. **Trends in system aspects of database management**, In Proc. Intl. Conf. on Databases (ICOD-2), Wiley & Sons 1983.

[UHW+09] Urbanski, Stephan; Huber, Eduard; Wieland, Matthias; Leymann, Frank; Nicklas, Daniela: **PerFlows for the Computers of the 21st Century**. In: Proceedings of the Seventh Annual IEEE Conference on Pervasive Computing and Communications : PerCom'09 Workshops, in 5th IEEE PerCom Workshop on Context Modeling and Reasoning; Galveston, 17.-21. March 2009

[VVK08] Jussi Vanhatalo, Hagen Völzer, Jana Koehler: **The Refined Process Structure Tree**. BPM 2008: 100-115

[WCL+05] Weerawarana, Sanjiva; Curbera, Francisco; Leymann, Frank; Storey, Tony; Ferguson, Donald F.: **Web Services Platform Architecture**, Prentice Hall, 2005.

[WR90] H. Wächter and A. Reuter. **Base concepts and realization strategies of the ConTract model** (in German), Informatik Forschung und Entwicklung 5, 202 – 212, 1990.

[WVKG08] Wang, T., Vonk, J., Kratz, B., Grefen, P.: **A survey on the history of transaction management: from flat to grid transactions**. Distributed and Parallel Databases 23(3) (2008) 235-270

[WWC92] G. Wiederhold, P. Wegner, and S. Ceri. **Towards Megaprogramming: A paradigm for component-based programming**, Comm. ACM 35(22) 1992, 89 – 99.

[X/O93] X/Open Guide, **Distributed Transaction Processing Reference Model** (Version 2), X/Open Company Ltd., U.K., 1993.

[ZBDH06] J. M. Zaha, A. Barros, M. Dumas, A. ter Hofstede, **Let's Dance: A Language for Service Behavior Modeling**, in: Proceedings 14th International Conference on Cooperative Information Systems (CoopIS), vol. 4275 of Lecture Notes in Computer Science, Springer Verlag, 2006.

[ZDH+08] Johannes Maria Zaha, Marlon Dumas, Arthur H.M. ter Hofstede, Alistair Barros and Gero Decker. **Bridging Global and Local Models of Service-oriented Systems**. IEEE Transactions on Systems, Man, and Cybernetics, Part C. Special Issue on Enterprise Services Computing and Industrial Applications. Vol. 38, Issue 3 (2008), pp. 302-31

A. Erfolgte und geplante Veröffentlichungen

A.1 Dissertationen

[Mas09] Peter Massuthe. **Operating Guidelines for Services**. Dissertation, Humboldt-Universität zu Berlin, Eindhoven University of Technology, April 2009. ISBN 978-90-386-1702-2.

A.2 Zeitschriftenbeiträge

[ALM+08] Wil M. P. van der Aalst, Niels Lohmann, Peter Massuthe, Christian Stahl, and Karsten Wolf. **Multiparty Contracts: Agreeing and Implementing Interorganizational Processes**. *The Computer Journal*, 2008. Note: (Accepted for publication).

[DKB08] Decker, Gero; Kopp, Oliver; Barros, Alistair: An **Introduction to Service Choreographies**. In: Information Technology. Vol. 50(2), 2008, Oldenbourg Verlag.

[DKLW09] Decker, Gero; Kopp, Oliver; Leymann, Frank; Weske, Mathias: **Interacting services: from specification to execution**. In: Data & Knowledge Engineering, Elsevier Science Publishers, 2009.

[KKL08a] Khalaf, Rania; Kopp, Oliver; Leymann, Frank: **Maintaining Data Dependencies Across BPEL Process Fragments**. In: International Journal of Cooperative Information Systems (IJCIS). Vol. 17(3), World Scientific, 2008.

[KL08] Kopp, Oliver; Leymann, Frank: **Choreography Design Using WS-BPEL**. In: Lohmet, David B. (Hrsg): Bulletin of the IEEE Computer Society Technical Committee on Data Engineering. Vol. 31(3), IEEE Computer Society Press, 2008.

[KMW07] Kathrin Kaschner, Peter Massuthe, and Karsten Wolf. **Symbolic Representation of Operating Guidelines for Services**. *Petri Net Newsletter*, 72: 21-28, April 2007.

[KMWL09] Kopp, Oliver; Martin, Daniel; Wutke, Daniel; Leymann, Frank: **The Difference Between Graph-Based and Block-Structured Business Process Modelling Languages**. In: Ulrich Frank (Hrsg): Enterprise Modelling and Information Systems. Vol. 4(1), Duisburg: Gesellschaft für Informatik e.V. (GI), 2009.

[LMSW08] Niels Lohmann, Peter Massuthe, Christian Stahl, and Daniela Weinberg. **Analyzing Interacting WS-BPEL Processes Using Flexible Model Generation**. *Data Knowl. Eng.*, 64(1): 38-54, January 2008.

[LVOS09] Niels Lohmann, H.M.W. Verbeek, Chun Ouyang, and Christian Stahl. **Comparing and Evaluating Petri Net Semantics for BPEL**. *Int. J. Business Process Integration and Management*, 4(1):60-73, 2009.

[MSSW08] Peter Massuthe, Alexander Serebrenik, Natalia Sidorova, and Karsten Wolf. **Can I find a Partner? Undecidablity of Partner Existence for Open Nets**. *Information Processing Letters*, 108(6): 374--378, November 2008.

[MW06] Peter Massuthe and Karsten Wolf. **Operating Guidelines for Services**. *Petri Net Newsletter*, 70: 9-14, April 2006.

[MW07] Peter Massuthe and Karsten Wolf. **An Algorithm for Matching Non-deterministic Services with Operating Guidelines**. *International Journal of Business Process Integration and Management (IJBPIM)*, 2(2): 81-90, 2007.

[RWB+07] Wolfgang Reisig, Karsten Wolf, Jan Bretschneider, Kathrin Kaschner, Niels Lohmann, Peter Massuthe, and Christian Stahl. **Challenges in a Service-Oriented World**. *ERCIM News*, 70: 28-29, July 2007.

[SMB09] Christian Stahl, Peter Massuthe, and Jan Bretschneider. **Deciding Substitutability of Services with Operating Guidelines**. *Transactions on Petri Nets and Other Models of Concurrency II, Special Issue on Concurrency in Process-Aware Information Systems*, 2(5460): 172-191, March 2009.

A.3 Konferenzbeiträge

[BKvLL09] Bischof, Marc; Kopp, Oliver; van Lessen, Tammo; Leymann, Frank: **BPELscript: A Simplified Script Syntax for WS-BPEL 2.0**. In: 2009 35th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2009)

[DKLPW08] Decker, Gero; Kopp, Oliver; Leymann, Frank; Pfitzner, Kerstin; Weske, Mathias: **Modeling Service Choreographies using BPMN and BPEL4Chor**. In: Proceedings of the 20th International Conference on Advanced Information Systems Engineering (CAiSE '08), 2008

[DKLW07] Decker, Gero; Kopp, Oliver; Leymann, Frank; Weske, Matthias: **BPEL4Chor: Extending BPEL for Modeling Choreographies**. In: IEEE Computer Society (Hrsg): Proceedings of the IEEE 2007 International Conference on Web Services (ICWS 2007), Salt Lake City, Utah, USA, July 2007

[KKL07a] Khalaf, Rania; Kopp, Oliver; Leymann, Frank: **Maintaining Data Dependencies Across BPEL Process Fragments**. In: Krämer, Bernd J. (Hrsg); Lin, Kwei-Jay (Hrsg); Narasimhan, Priya (Hrsg): Service-Oriented Computing - ICSOC 2007

[KKL08b] Kopp, Oliver; Khalaf, Rania; Leymann, Frank: **Deriving Explicit Data Links in WS-BPEL Processes**. In: Proceedings of the International Conference on Services Computing, Industry Track, SCC 2008

[KLM+08] Dieter König, Niels Lohmann, Simon Moser, Christian Stahl, and Karsten Wolf. **Extending the Compatibility Notion for Abstract WS-BPEL Processes**. In Wei-Ying Ma, Andrew Tomkins, and Xiaodong Zhang, editors, Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008, pages 785-794, April 2008. ACM.

[KML09] Kopp, Oliver; Mietzner, Ralph; Leymann, Frank: **The Influence of an External Transaction on a BPEL Scope**. CoopIS 2009. Springer.

[KMWL08] Kopp, Oliver; Martin, Daniel; Wutke, Daniel; Leymann, Frank: **On the Choice Between Graph-Based and Block-Structured Business Process Modeling Languages**. In: Modellierung betrieblicher Informationssysteme (MobIS 2008). Saarbrücken, Germany, November 27 - 28, 2008

[LMSW06] Niels Lohmann, Peter Massuthe, Christian Stahl, and Daniela Weinberg. **Analyzing Interacting BPEL Processes**. In Schahram Dustdar, José Luiz Fiadeiro, and Amit Sheth, editors, *Business Process Management, 4th International Conference, BPM 2006, Vienna, Austria, September 5-7, 2006, Proceedings*, volume 4102 of *Lecture Notes in Computer Science*, pages 17-32, September 2006. Springer-Verlag.

[LMW07a] Niels Lohmann, Peter Massuthe, and Karsten Wolf. **Operating Guidelines for Finite-State Services**. In Jetty Kleijn and Alex Yakovlev, editors, *28th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency, ICATPN 2007, Siedlce, Poland, June 25-29, 2007, Proceedings*, volume 4546 of *Lecture Notes in Computer Science*, pages 321-341, 2007. Springer-Verlag.

[LMW07b] Niels Lohmann, Peter Massuthe, and Karsten Wolf. **Behavioral Constraints for Services**. In Gustavo Alonso, Peter Dadam, and Michael Rosemann, editors, *Business Process Management, 5th International Conference, BPM 2007, Brisbane, Australia, September 24-28, 2007, Proceedings*, volume 4714 of *Lecture Notes in Computer Science*, pages 271-287, September 2007. Springer-Verlag.

[RBF+07] Wolfgang Reisig, Jan Bretschneider, Dirk Fahland, Niels Lohmann, Peter Massuthe, and Christian Stahl. **Services as a Paradigm of Computation**. In Cliff B. Jones, Zhiming Liu, and Jim Woodcock, editors, *Formal Methods and Hybrid Real-Time Systems, Essays in Honor of Dines Bjorner and Chaochen Zhou on the Occasion of Their 70th Birthdays, Papers presented at a Symposium held in Macao, China, September 24-25, 2007*, volume 4700 of *Lecture Notes in Computer Science*, pages 521-538, September 2007. Springer-Verlag.

[RFL+06] Wolfgang Reisig, Dirk Fahland, Niels Lohmann, Peter Massuthe, Christian Stahl, Daniela Weinberg, Karsten Wolf, and Kathrin Kaschner. **Analysis Techniques for Service Models**. In *Second International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, 2006 (ISoLA 2006), 15-19 November 2006, Paphos, Cyprus*, pages 11-17, November 2006. IEEE Computer Society

[WMKL09] Wieland, Matthias; Martin, Daniel; Kopp, Oliver; Leymann, Frank: **SOEDA: A Methodology for Specification and Implementation of Applications on a Service-Oriented Event-Driven Architecture**. In: Proceedings of the 12th International Conference on Business Information Systems (BIS 2009). Poznan, Poland April 27-29, 2009

A.4 Workshopbeiträge

[ALM+07] Wil M. P. van der Aalst, Niels Lohmann, Peter Massuthe, Christian Stahl, and Karsten Wolf. From Public Views to Private Views -- Correctness-by-Design for Services. In

Marlon Dumas and Reiko Heckel, editors, Web Services and Formal Methods, Forth International Workshop, WS-FM 2007 Brisbane, Australia, September 28-29, 2007, Proceedings, volume 4937 of Lecture Notes in Computer Science, pages 139-153, 2008. Springer-Verlag.

[DKP07] Decker, Gero; Kopp, Oliver; Puhlmann, Frank: **Service Referrals in BPEL-based Choreographies**. In: Gorton, Stephen (Hrsg); Solanki, Monika (Hrsg); Reiff-Marganiec, Stephen (Hrsg): Proceedings of the 2nd European Young Researchers Workshop on Service Oriented Computing (YR-SOC 2007)

[KELU07] Kopp, Oliver; Eberle, Hanna; Leymann, Frank; Unger, Tobias: **From Process Models to Business Landscapes**. In: Nüttgens, Markus (Hrsg); Rump, Frank J. Rump (Hrsg); Gadatsch, Andreas (Hrsg): EPK 2007: Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten

[KL09] Kopp, Oliver; Leymann, Frank: **Do We Need Internal Behavior in Choreography Models?**. Proceedings of the 1st Central-European Workshop on Services and their Composition, ZEUS 2009, Stuttgart, Germany, March 2-3, 2009

[KFL06] Oliver Kopp, Carsten Frenkler, and Niels Lohmann. **Korrektheit und Zuverlässigkeit zusammengesetzter Web Services am Beispiel der Geschäftsprozess-Modellierungssprache BPEL**. In *Forschungsoffensive ''Software Engineering 2006'', Statuskonferenz, 26.-28. Juni 2006*, July 2006. Bundesministerium für Bildung und Forschung (BMBF).

[KMW06] Kathrin Kaschner, Peter Massuthe, and Karsten Wolf. **Symbolische Repräsentation von Bedienungsanleitungen für Services**. In Daniel Moldt, editor, *13. Workshop Algorithmen und Werkzeuge für Petrinetze (AWPN 2006), Proceedings*, pages 54-61, September 2006. Universität Hamburg. Note: In German.

[KvLN08] Kopp, Oliver; van Lessen, Tammo; Nitzsche, Jörg: **The Need for a Choreography-aware Service Bus**. In: YR-SOC 2008

[KUL06] Kopp, Oliver; Unger, Tobias; Leymann, Frank: **Nautilus Event-driven Process Chains: Syntax, Semantics, and their mapping to BPEL**. In: Nüttgens, M. (Hrsg); Rump, F.J. (Hrsg); Mendling, J. (Hrsg): Proceedings of the 5th GI Workshop on Event-Driven Process Chains (EPK 2006)

[KWL09a] Kopp, Oliver; Wieland, Matthias; Leymann, Frank: **Towards Choreography Transactions**. Proceedings of the 1st Central-European Workshop on Services and their Composition, ZEUS 2009, Stuttgart, Germany, March 2-3, 2009

[KWL09b] Kopp, Oliver; Wieland, Matthias; Leymann, Frank: **External and Internal Events in EPCs: e2EPCs**. In: 2nd International Workshop on Event-Driven Business Process Management (edBPM09), 2009.

[KWM+08a] Kopp, Oliver; Wetzstein, Branimir; Mietzner, Ralph; Pottinger, Stefan; Karastoyanova, Dimka; Leymann, Frank: **A Model-Driven Approach to Implementing Coordination Protocols in BPEL**. In: 1st International Workshop on Model-Driven Engineering for Business Process Management (MDE4BPM 2008)

[LK08] Lohmann, Niels; Kopp, Oliver: **Tools4BPEL4Chor**. In: YR-SOC 2008

[LKLR08] Lohmann, Niels; Kopp, Oliver; Leymann, Frank; Reisig, Wolfgang: Analyzing **BPEL4Chor: Verification and Participant Synthesis**. In: Dumas, Marlon (Hrsg); Heckel, Reiko (Hrsg): Web Services and Formal Methods, Forth International Workshop, WS-FM 2007 Brisbane, Australia, volume 4937 of *Lecture Notes in Computer Science*, pages 46-60, April 2008. Springer-Verlag

[Loh08a] Niels Lohmann. **A Feature-Complete Petri Net Semantics for WS-BPEL 2.0**. In Marlon Dumas and Reiko Heckel, editors, *Web Services and Formal Methods, Forth International Workshop, WS-FM 2007, Brisbane, Australia, September 28-29, 2007, Proceedings*, volume 4937 of *Lecture Notes in Computer Science*, pages 77-91, April 2008. Springer-Verlag.

[Loh08b] Niels Lohmann. **A Feature-Complete Petri Net Semantics for WS-BPEL 2.0**. In Kees van Hee, Wolfgang Reisig, and Karsten Wolf, editors, *Proceedings of the Workshop on Formal Approaches to Business Processes and Web Services (FABPWS'07)*, pages 21-35, June 2007. University of Podlasie

[MW08] Peter Massuthe and Daniela Weinberg. **Fiona: A Tool to Analyze Interacting Open Nets**. In Niels Lohmann and Karsten Wolf, editors, *Proceedings of the 15th German Workshop on Algorithms and Tools for Petri Nets, AWPN 2008, Rostock, Germany, September 26--27, 2008*, volume 380 of *CEUR Workshop Proceedings*, pages 99-104, September 2008. CEUR-WS.org.

[PDKL07] Pfitzner, Kerstin; Decker, Gero; Kopp, Oliver; Leymann, Frank: **Web Service Choreography Configurations for BPMN**. Proceedings of the 3rd International Workshop on Engineering Service-Oriented Application (WESOA'2007)

A.5 Technische Berichte

[KKL07b] Kopp, Oliver; Khalaf, Rania; Leymann, Frank: **Reaching Definitions Analysis Respecting Dead Path Elimination Semantics in BPEL Processes**, Technischer Bericht Nr. 2007/04

[KML08] Kopp, Oliver; Mietzner, Ralph; Leymann, Frank: **Abstract Syntax of WS-BPEL 2.0**, Technischer Bericht Nr. 2008/06

[KWM+08b] Kopp, Oliver; Wetzstein, Branimir; Mietzner, Ralph; Unger, Tobias; Pottinger, Stefan; Karastoyanova, Dimka; Michael, Sabine; Leymann, Frank: **A Model-Driven Approach to Implementing Coordination Protocols in BPEL**, Technischer Bericht Nr. 2008/02

[LVO+07] Niels Lohmann, H. M. W. Verbeek, Chun Ouyang, Christian Stahl, and Wil M. P. van der Aalst. **Comparing and Evaluating Petri Net Semantics for BPEL**. Computer Science Report 07/23, Technische Universiteit Eindhoven, The Netherlands, August 2007.

[Loh07] Niels Lohmann. **A Feature-Complete Petri Net Semantics for WS-BPEL 2.0 and its Compiler BPEL2oWFN**. Informatik-Berichte 212, Humboldt-Universität zu Berlin, Berlin, Germany, August 2007.

[LMW06] Niels Lohmann, Peter Massuthe, and Karsten Wolf. **Operating Guidelines for Finite-State Services**. Informatik-Berichte 210, Humboldt-Universität zu Berlin, December 2006.

[RKDL08] Reimann, Peter; Kopp, Oliver; Decker, Gero; Leymann, Frank: **Generating WS-BPEL 2.0 Processes from a Grounded BPEL4Chor Choreography**, Technischer Bericht Nr. 2008/07

[WMKL08] Wieland, Matthias; Martin, Daniel; Kopp, Oliver; Leymann, Frank: **Events Make Workflows Really Useful**, Technischer Bericht Nr. 2008/09

A.6 Geplant

Die folgenden Veröffentlichungen fassen Teilergebnisse des Projekts zusammen.

[Kop10] Dissertation über Partnerübergreifende Prozesse, deren Modellierung, Fehlerbehandlung und Transaktionen (Kopp, Oliver)

[Loh09] Lohmann, Niels. **Correctness of services and their composition**. Dissertation Universität Rostock

[UEKL09] Papier zum Unterprozessspektrum und Autonomiegraden (Unger, Tobias; Eberle, Hanna; Kopp, Oliver; Leymann, Frank)