

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



Oliver Kopp
Universität Stuttgart
Universitätsstraße 38
70569 Stuttgart

Carsten Frenkler
Gedilan Technologies GmbH
Mainzer Str. 20
14612 Falkensee

Niels Lohmann
Humboldt-Universität zu Berlin
Unter den Linden 6
10099 Berlin

Kurzfassung

Unternehmensübergreifende Geschäftsprozesse werden zunehmend nach dem Paradigma der Services organisiert. Dabei stellen sich Fragen nach der Komponierbarkeit, Bedienbarkeit, Austauschbarkeit sowie der Rücksetzbarkeit (Kompensation) im Fehlerfall. In diesem Vorhaben werden Methoden und Werkzeuge zum Umgang mit solchen Fragen entwickelt, am Beispiel der Geschäftsprozess-Modellierungssprache BPEL erprobt und in ein Entwurfswerkzeug der Firma Gedilan Technologies integriert.

1. Einleitung und Vorstellung des Themenkomplexes

Geschäftsprozesse werden in zunehmendem Maße informationstechnisch als Workflows realisiert. Diese Entwicklung betraf zunächst hauptsächlich unternehmensinterne Prozesse. Angetrieben durch die fortschreitende Globalisierung, die Idee virtueller Unternehmen und dank leistungsfähiger Kommunikationsmöglichkeiten über das Internet gewinnen unternehmensübergreifende Geschäftsprozesse (im weiteren Verlauf als „partnerübergreifend“ bezeichnet) zunehmend an Bedeutung. Solche Prozesse können in der Regel nicht als herkömmliche Workflows abgebildet werden, weil dies z.B. die unerwünschte Preisgabe von Firmengeheimnissen beteiligter Partner zur Folge hätte.

In der Praxis werden gegenwärtig Konzepte für die informationstechnische Realisierung partnerübergreifender Geschäftsprozesse entwickelt. Im Mittelpunkt dieser Konzepte steht der Begriff des Services. Ein Service ist eine Menge von Aktivitäten, angeordnet in einer Kontrollstruktur, z.B. einem Workflow. Ein Service enthält zudem eine Schnittstelle und einen Bezeichner. Für die Beschreibung der Schnittstelle eines Services hat sich die Sprache WSDL [CM+05] als Standard etabliert, für die Bezeichnung und Lokalisierung UDDI [CHvRR04]. Für die Beschreibung der Kontrollstruktur eines Services wird die Sprache BPEL (Business Process Execution Language for Web Services, BPEL4WS, WS-BPEL) [CGK+02] verwendet. BPEL4WS 1.1 wird bereits vielen bedeutenden Firmen weltweit unterstützt. Die Standardisierung der Version 2.0 wird derzeit von OASIS vorangetrieben [OASISTC].

Partnerübergreifende Geschäftsprozesse werden aus Services zusammengesetzt und bilden ihrerseits einen partnerübergreifenden Service. Dies geschieht durch eine (zwischen den beteiligten Partnern auszuhandelnde) Service-Choreografie [KBR04] oder im Rahmen einer Service-basierten Architektur (SOA) [Got00].

Angesichts der ökonomischen Relevanz partnerübergreifender Geschäftsprozesse ist die Korrektheit und Zuverlässigkeit ihrer informationstechnischen Realisierungen von herausragender Bedeutung. Zur Sicherung der Qualität sind einerseits Konzepte nötig, mit deren Hilfe alle Aspekte der Interaktion zwischen Services adäquat spezifiziert werden können. Gegenwärtige Konzepte genügen diesen Anforderungen noch nicht in ausreichendem Umfang. Andererseits braucht man formale Methoden und Werkzeuge, mit deren Hilfe Fragen über die Korrektheit von Services und deren Interaktion beantwortet werden können. Speziell auf Services zugeschnittene Methoden sind derzeit nicht verfügbar.

Aus diesen Sachverhalten folgt die Aufgabenstellung, Konzepte für die strukturierte Spezifikation wichtiger Aspekte der Interaktion von Services bereitzustellen sowie spezifische formale Methoden und Werkzeuge für die Sicherung korrekten Verhaltens von Services zu entwickeln.

Für Spezifikationen, die in der Sprache BPEL gegeben sind, werden in diesem Projekt entsprechende Konzepte, Methoden und Werkzeugkomponenten entwickelt, prototypisch in Werkzeuge zum Geschäftsprozess-Management integriert und an realen Anwendungsfällen erprobt.

1.1 Korrektheit von Services

Petrinetze sind eine anerkannte formale Beschreibung von Services [RSS05]. In diesem Projekt wird eine Infrastruktur aufgebaut, mit deren Hilfe verfügbare Verifikationswerkzeuge für Petrinetze für BPEL-Spezifikationen anwendbar gemacht werden können. Zu diesem Zweck wird eine Übersetzung von BPEL-Prozessen in Petrinetze implementiert. Die entstehenden Netze dienen dann als Eingabe für Model-Checking-Werkzeuge und andere Petrinetz-basierte Werkzeuge. Die generierten Petrinetze sollen dabei so klein wie möglich gehalten werden, damit die nachgeschalteten Analysewerkzeuge auch für BPEL-Prozesse realistischer Größen Ergebnisse liefern. Zu diesem Zweck entwickeln wir eine Technologie für eine flexible Modellbildung, die die Generierung des Petrinetzmodells aus möglichst abstrakten Bausteinen vorsieht. Diese Bausteine werden in einer erweiterbaren Bibliothek verwaltet. Ihre Anwendbarkeit in einem bestimmten Kontext wird durch statische Analyse auf dem BPEL-Quelltext überprüft.

Das Konzept flexibler Modellbildung stellt eine innovative Herangehensweise an die Generierung formaler Modelle aus realen Spezifikations Sprachen dar. Das wissenschaftliche Ziel besteht in der Demonstration der Möglichkeit, mit statischen Analysetechniken und einer geeigneten Bibliothek elementarer Petrinetzbausteine ein kompaktes Modell aus einer Spezifikation realistischer Größe zu generieren, das Analysewerkzeuge beherrschen können.

Das technische Ziel besteht in der Entwicklung eines leistungsfähigen Übersetzers BPEL2oWFN von BPEL in offene Workflow-Netze (oWFNs). Dieser Übersetzer soll aus einer gegebenen Eigenschaft und einem BPEL-Prozess ein möglichst kleines Petrinetzmodell generieren, das die Entscheidung der Eigenschaft für den Originalprozess gestattet. Typische solche Eigenschaften sind die Bedienbarkeit, das externe Kommunikationsverhalten (für die Betrachtung von Austauschbarkeit und Konsistenz) sowie Model Checking Service-interner Eigenschaften. Die Petrinetze werden im standardisierten Format PNML [Bea03] abgelegt und sind dadurch für alle Petrinetz-basierten Werkzeuge verfügbar. Über die Plattform „Model Checking Kit“ [MCK] ist beispielsweise der Modelchecker „Spin“ [Hol03] zum Model Checking unmittelbar verwendbar.

1.2 Eigenschaften von Services

In diesem Projektschwerpunkt werden Service-spezifische Varianten für Zuverlässigkeitskriterien untersucht. Im Mittelpunkt steht dabei die Sicherstellung der Abwesenheit von Deadlocks bei der Interaktion von Services. Das Deadlock-Problem kann als paradigmatisch für Probleme der Interaktion zwischen Services angesehen werden. Lösungen für dieses Problem sind mit hoher Wahrscheinlichkeit auch auf andere Probleme übertragbar. Folgende Varianten werden dabei berücksichtigt:

1. *Konsistenz.* Eine Menge von Services ist konsistent, wenn sie deadlockfrei miteinander kommunizieren. Konsistenz eines partnerübergreifenden Geschäftsprozesses kann nur dann überprüft werden, wenn entweder die Partner bereit sind, die Kontrollstruktur ihres Services zu veröffentlichen oder die Konsistenz anhand abstrakter Sichten (siehe unten) der beteiligten Services überprüft wird.
2. *Bedienbarkeit.* Ein Service S ist bedienbar, wenn es Services gibt, mit denen S deadlockfrei interagiert. Bedienbarkeit ist eine Eigenschaft eines einzelnen Service S und kann vom Entwickler von S allein überprüft werden. Bedienbarkeit ist eine Minimalanforderung für ein vernünftiges Design eines Services.
3. *Austauschbarkeit.* Ein Service S_1 ist gegen einen anderen Service S_2 austauschbar, falls jede konsistente Menge von Services konsistent bleibt, wenn man in ihr S_1 durch S_2 ersetzt. Austauschbarkeit betrifft typischerweise zwei Services ein und desselben Partners und kann daher von diesem allein überprüft werden.
4. *Abstrakte Sichten.* Eine abstrakte Sicht eines Services gibt einige Aspekte der Kontrollstruktur wieder (z.B. die Kommunikationsaktivitäten) bei gleichzeitiger Abstraktion von anderen Aspekten. Sie dient der Dokumentation eines Services im eigenen Unternehmen oder zur Übermittlung von Information über den Service an andere Partner. Für letzteren Zweck muss der Abstraktionsgrad hoch genug sein, um die den Service betreffenden Firmengeheimnisse zu wahren.
5. *Bedienungsanleitungen.* Als Alternative zu abstrakten Sichten wird die automatische Generierung von Bedienungsanleitungen untersucht. Eine Bedienungsanleitung für einen Service beschreibt, wie Partner des Services deadlockfrei mit dem Service kommunizieren können.

Diese Eigenschaften werden unabhängig von konkreten Spezifikationssprachen betrachtet. Stattdessen werden sie auf der Basis des theoretischen Modells der offenen Workflow-Netze, einer Klasse von Petrinetzen (siehe [RSS05]), studiert. Dies sichert die Langlebigkeit und Allgemeingültigkeit der Resultate.

1.3 Komposition von Services

Für die Zuverlässigkeit unternehmensübergreifender Geschäftsprozesse ist es unerlässlich, sie für ein korrektes Zusammenwirken mehrerer Services notwendigen Spezifikationen adäquat formulieren zu können. Zu diesen Spezifikationen zählen:

1. Die Beschreibung der Kompositionsstruktur: Wie werden die beteiligten Services miteinander kombiniert?
2. Die Beschreibung von Autonomie: Welche Freiheitsgrade für selbständige Änderungen hat ein beteiligter Partner?
3. Die Beschreibung transaktionaler Zusammenhänge: Wie kann sichergestellt werden, dass ein partnerübergreifender Prozess bei allen beteiligten Partnern konsistente Zustände hinterlässt?
4. Die Beschreibung einer Fehlerkoordination: Wie kann durch ein Zusammenwirken aller Partner auf Fehler und Ausnahmen reagiert werden?

Firmen beschreiben ihre Geschäftsprozesse fast immer aus ihrer eigenen Sicht, d.h. firmenzentrisch. In notwendigen Interaktionen mit Partnern werden diese als „opak“ aufgefasst, d.h., es ist insbesondere nicht bekannt, ob die Partner von sich aus die Interaktionen wieder als Geschäftsprozesse realisieren. Durch die allgemeine Akzeptanz von BPEL als Beschreibungssprache von Prozessen, die mit Partnern interagieren, ist es realistisch anzunehmen, dass jeder beteiligte Partner seine Interaktionen mit den anderen Partnern als einen Geschäftsprozess in BPEL beschreibt. Dadurch können Interaktionen zwischen Partnern als partnerübergreifende Geschäftsprozesse aufgefasst werden.

Solche partnerübergreifende Geschäftsprozesse können grundsätzlich auf zwei verschiedene Arten beschrieben werden, nämlich hierarchisch oder ebenbürtig („peer-to-peer“). In einem partnerübergreifenden Prozess mit mehreren Partnern werden im Allgemeinen beide Arten gleichzeitig vorkommen. Die Sprache BPEL reicht für solche Beschreibungen nicht aus und wird entsprechend erweitert werden.

Bei der Komposition mehrerer firmenzentrischer Prozesse in eine neue Einheit, einen partnerübergreifenden Geschäftsprozess, müssen die Partner teilweise die Autonomie ihrer individuellen Geschäftsprozesse aufgeben. Die Entwicklung von Mechanismen, mit denen Anforderungen und die Bereitschaft zur Aufgabe von Autonomie spezifiziert werden und auch abgeglichen werden kann, ist ein Ziel des Vorhabens.

Während Fehlerbehandlung und Transaktionskonzepte in einzelnen Geschäftsprozessen durch BPEL beschrieben werden, ist sowohl eine Fehlerbehandlung in partnerübergreifenden Geschäftsprozessen als auch ein entsprechender Transaktionsmechanismus noch zu erforschen: Dies ist ein weiteres Ziel des Vorhabens.

1.4 Validierung, Produktreifung und Integration

Die in diesem Vorhaben entwickelten Technologien messen sich an den Erfordernissen der industriellen Praxis. Zu diesem Zweck werden sämtliche Entwicklungen mit praxisnahen Fallbeispielen begleitet. Die Werkzeuge werden in Nautilus [Ged06b], einer in zahlreichen Firmenanwendungen genutzte Software zur Geschäftsprozessmodellierung, integriert und somit unter realen Bedingungen validiert. Bestandteil der Integration ist ein Ex- und Importer zwischen BPEL und der derzeit in Nautilus verwendeten Modellierungssprache, die auf den ereignisgesteuerten Prozessketten (EPKs) basiert.

Eine der Aufgaben der Firma Gedilan Technologies innerhalb des Projekts Tools4BPEL besteht darin, die von den Projektpartnern realisierten Prototypen in die Nautilus-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 Nautilus angebunden werden.

2. Projektstatus

Am Anfang des Projekts stand die Arbeit an Nautilus und der offenen Workflownetzen im Vordergrund. In Nautilus wurden sowohl ein BPEL-Importer als auch ein BPEL-Exporter prototypisch implementiert. Der BPEL-Exporter wurde zur Produktreife entwickelt und kann bereits erworben werden [Ged06a]. Die Ausgabe des Exporters kann als Eingabe für das Analysewerkzeug „BPEL2oWFN“ verwendet werden. BPEL2oWFN ist ein Opensource-Werkzeug, das BPEL in ein offenes Workflownetz (oWFN) übersetzt und kann unter [T4B] bezogen werden. Ein oWFN ist eine Klasse von Petrinetzen, die auf die Analyse von Services zugeschnitten ist. In [HSS05] wurde eine Petrinetzsemantik für BPEL vorgestellt. Diese wird als Basis für die Übersetzung von BPEL in oWFNs verwendet. Die in [HSS05] vorgestellte Petrinetzsemantik deckt sowohl den positiven als auch den negativen Kontrollfluss¹ ab. Damit ist die Semantik im Gegensatz zu anderen Ansätzen, wie

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

vollständig. Die Vollständigkeit wird durch viele Stellen in den Petrinetzmustern erreicht. Jede BPEL-Aktivität wird in ein Petrinetzmuster abgebildet, die untereinander durch Transitionen verbunden werden. Abbildung 1 zeigt ein Petrinetzmuster für ein „invoke“.

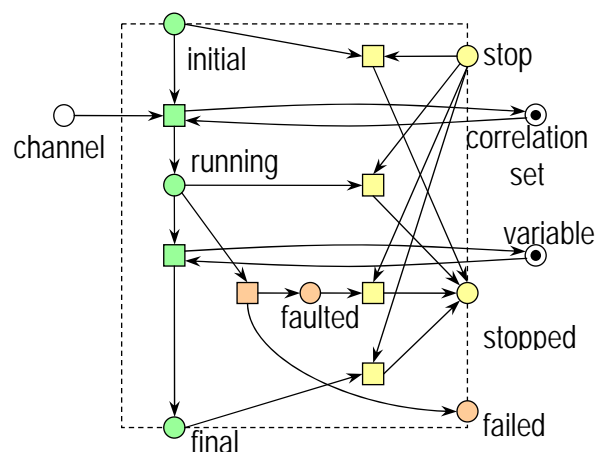


Abbildung 1: Petrinetzmuster für ein „invoke“

Die Verbindung mit anderen Aktivitäten wird über die Stellen „initial“ und „final“ vollzogen. Die Einbindung in den negativen Kontrollfluss erfolgt durch die Stellen „failed“ und „stop“. Um die Komplexität des entstehenden Petrinetzes zu reduzieren, wird das Konzept der „flexiblen Modellierung“ eingeführt, bei der bestimmte Stellen und Transitionen abhängig vom Anwendungsfall weggelassen werden. Soll beispielsweise nur der positive Kontrollfluss ohne Variablen und Prozessinteraktion modelliert werden, so kann der in Abbildung 2 gezeigte Baustein verwendet werden.

¹ Mit „negativer Kontrollfluss“ wird das Verhalten im Fehler- und Ausnahmefall, insbesondere das Zurücksetzen (Kompensieren) von Aktivitäten bezeichnet.

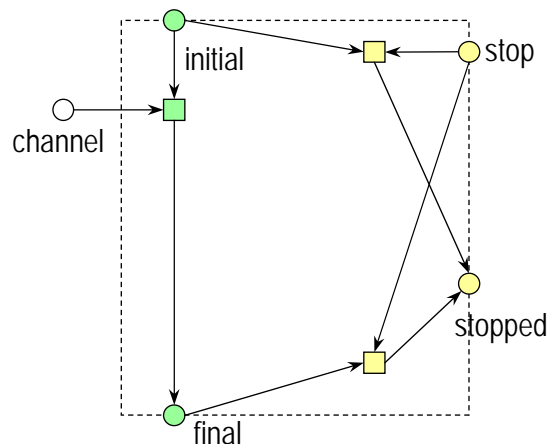


Abbildung 2: Petrinetzmuster für den positiven Kontrollfluss

Um die Konzepte der Bedienbarkeit und Bedienungsanleitungen zu demonstrieren, wurde auf der Fourth International Conference on Business Process Management (BPM2006) ein Papier eingereicht [LMS+06]. In ihm wurde ein Onlineshop vorgestellt. Abbildung 3 zeigt ihn als ereignisgesteuerte Prozesskette.

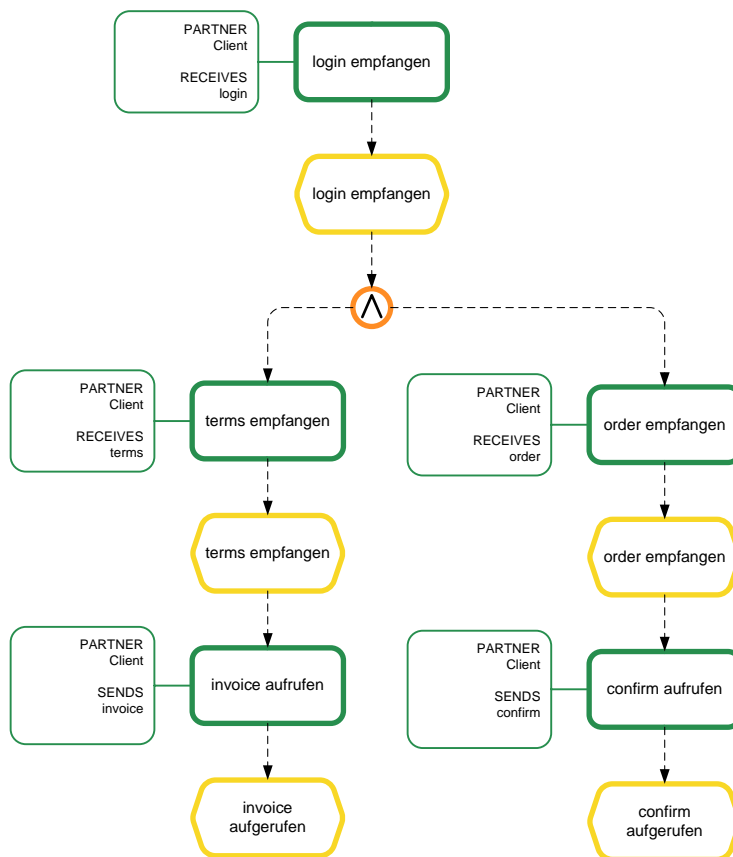


Abbildung 3: Die Nautilus-EPK des Onlineshops

Der Onlineshop bietet folgende Interaktion: Ein Kunde kann sich am Service einloggen, indem er die Nachricht „login“ schickt. Durch die Nachricht „order“ übermittelt der Kunde seine Bestellung und bestätigt mittels der Nachricht „terms“ die AGBs. Das System sendet die Rechnung („invoice“) und bestätigt die Bestellung („confirm“).

Die EPK des Onlineshops kann mittels des BPEL-Exporters als BPEL-Datei exportiert werden. Die Abbildung der Nautilus-EPKs von und nach BPEL unterscheidet sich von der in [Fö05] und [MLZ06] vorgestellten, da das Metamodell von Nautilus keine alternierende Folge von Funktionen und Ereignissen vorschreibt. Im Metamodell von Nautilus können beliebig viele Ereignisse vor, zwischen und nach Funktionen auftreten. Ereignisse müssen, ähnlich wie Funktionen und Ereignisse in dem in [NR02] gezeigten EPK-Metamodell, mittels Operatoren verbunden sein. Die Verwendung von mehreren Ereignissen zwischen Funktionen erlaubt dem Modellierer, detaillierte Zusammenhänge zwischen Funktionen zu beschreiben und dem BPEL-Exporter, die Ereignisse direkt in BPEL-transition conditions umzusetzen. Näheres zu dem Nautilus-Metamodell und der Übersetzung nach BPEL findet sich in [Ged06b].

BPEL2oWFN kann die exportierte Datei als Eingabe erhalten. Das resultierende offene Workflownetz ist in Abbildung 4 vereinfacht dargestellt.

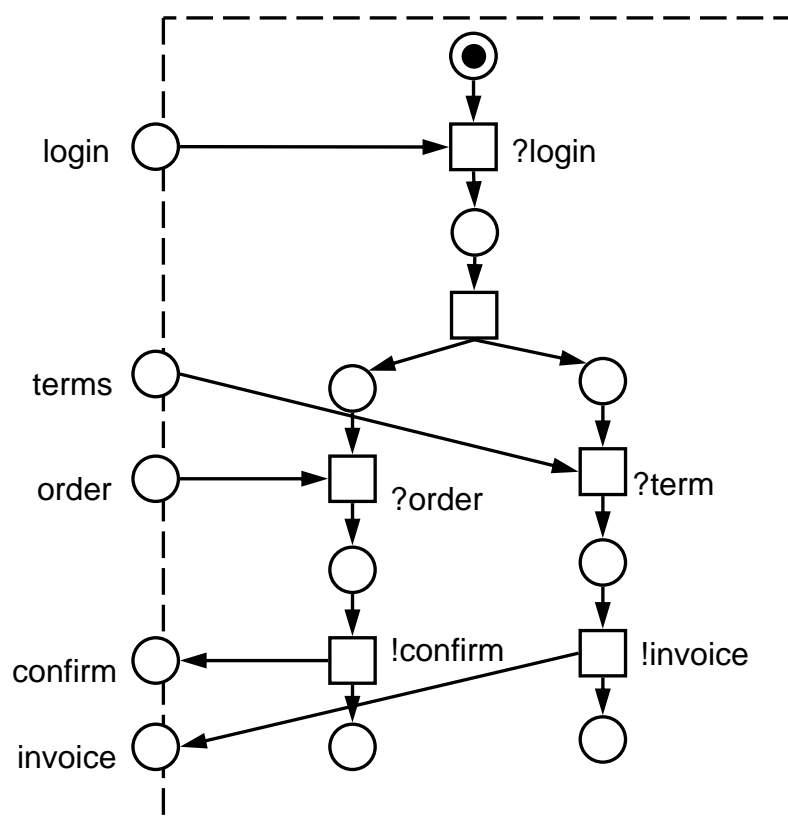


Abbildung 4: Das oWFN des Onlineshops

Ein „?“ modelliert das Senden einer Nachricht, ein „!“ das Empfangen einer Nachricht. Neben der Ausgabe in ein oWFN beherrscht BPEL2oWFN auch die Ausgabe in Petrinetze. Diese können dazu verwendet werden, um bereits vorhandene Modelchecker einzusetzen. Die owFN-

Ausgabe kann als Eingabe für das Analysewerkzeug Fiona (auch unter [T4B] erhältlich) dienen (vgl. Abbildung 5).

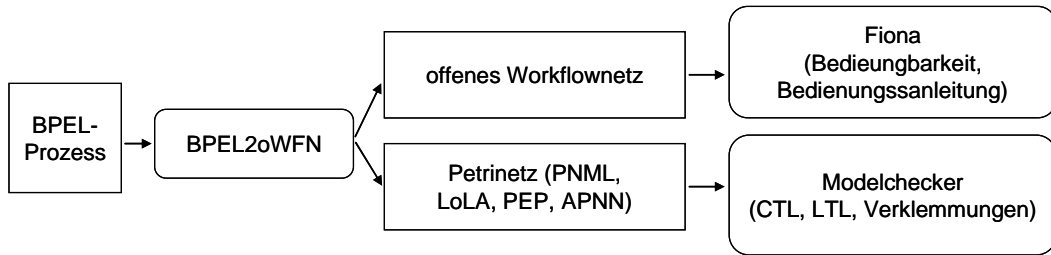


Abbildung 5: Möglichkeiten von BPEL2OWFN

Fiona kann die Bedienbarkeit eines oWFN entscheiden und falls Bedienbarkeit vorliegt, die zu dem oWFN gehörende Bedienungsanleitungen bestimmen. Eine Bedienungsanleitung ist ein Graph, bei denen die Knoten Zustände und die Kanten die Übergänge beschreiben. In jedem Knoten sind die Nachrichten enthalten, die von dem Partner gesendet (!Nachricht) oder empfangen (?Nachricht) werden können. In Abbildung 6 ist die Bedienungsanleitung für den Onlineshop gezeigt. Diese Bedienungsanleitung spiegelt das Verhalten aller möglichen bedienenden Partner wider. Jeder Partner, der einen beliebigen Pfad von der Wurzel zu dem Blatt des Graphen geht, ist ein bedienender Partner.

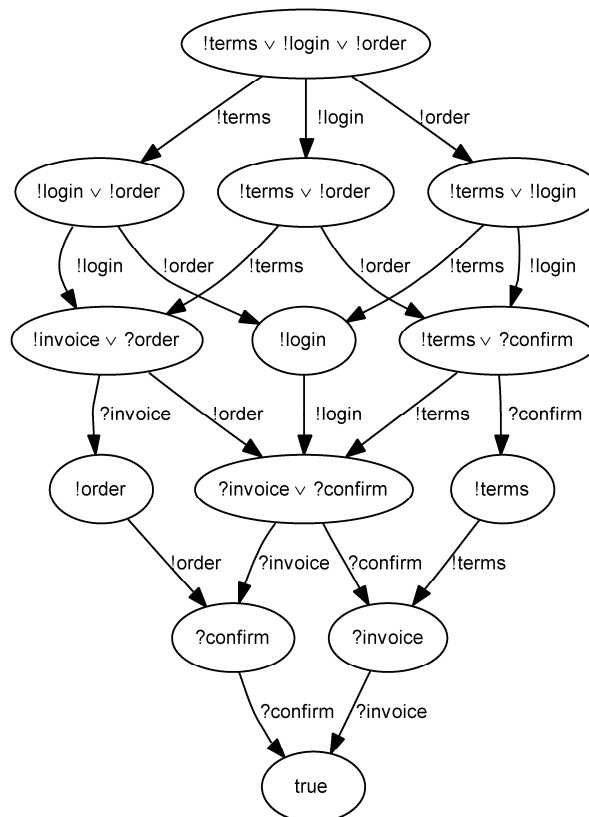


Abbildung 6: Die Bedienungsanleitung des Onlineshops

Durch Bedienungsanleitungen kann das Auffinden von Services vereinfacht werden. In einem Service-Broker wird die Bedienungsanleitung eines veröffentlichten Services bereitgestellt. Falls ein Partner einen Onlineshop sucht, so stellt er seine gewünschte Kommunikation als Suchanfrage an den Broker. Dieser sucht daraufhin in allen verfügbaren Bedienungsanleitungen, ob diese Anfrage als Teilgraph enthalten ist. Als weitere Bedingung muss der Teilgraph die Wurzel und das Blatt der Bedienungsanleitung enthalten, da sonst keine Kommunikation mit dem Service möglich ist. Durch Bedienungsanleitungen kann weiterhin die in [Mar04] definierte „Austauschbarkeit“ bestimmt werden. Weitere Ausführungen zu Bedienungsanleitungen finden sich in [RSS05], [MSS05], [MS05], sowie [Sch05].

3. Erfahrungen, Bewertungen

Auf der Statuskonferenz in Leipzig fand reger Austausch mit anderen Projekten statt. Insbesondere über die Komplexität der Petrinetze und das Kriterium der Austauschbarkeit. Das zeigt uns, dass die Themen auch für andere Projekte interessant sind. Für regen Erfahrungsaustausch sorgte auch die Abbildung der ereignisgesteuerten Prozessketten (EPKs) auf BPEL, da die EPKs ein anderes Ziel als BPEL besitzen.

Die Kunden von Gedilan zeigen Interesse an dem BPEL-Exporter. Weitere Erfahrungen liegen in diesem Bereich noch nicht vor, da der Exporter erst kurze Zeit am Markt verfügbar ist.

4. Ausblick

Die BPEL2oWFN-Schnittstelle ist als Bestandteil anstehender Major Releases von Nautilus vorgesehen. Ausgehend von dieser Schnittstelle kann das Werkzeug Fiona, über BPEL2oWFN, unmittelbar für in Nautilus modellierte Prozesse verwendet werden. Durch die enge Kooperation mit den universitären Partnern über Fallbeispiele und Validierung werden die von den universitären Partnern entwickelten Werkzeuge den Anforderungen für die Praxis genügen.

Neben der Integration in Nautilus können die Werkzeuge BPEL2oWFN und Fiona für die Verifikation von beliebigen BPEL-Prozessmodellen verwendet werden. Somit kann jeder BPEL-Prozess vor seiner Ausführung auf Fehler überprüft werden.

Mit der Entwicklung von Konzepten zur Modellierung von verteilten Geschäftsprozessen wurde gerade erst begonnen. Wir hoffen durch unsere Konzepte, die derzeitig verfügbare, doch nicht etablierte, Beschreibungssprache für Choreografie, WS-CDL [KBR04], zu ergänzen. Weiterhin gehen wir davon aus, durch unsere Ideen auch die Fortentwicklung von WS-CDL zu beeinflussen.

Literaturverzeichnis

- [Bea03] J. Billington et al. The Petri Net Markup Language: Concepts, Technology, and Tools, 2003.
- [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/>
- [CHvRR04] L. Clement, A. Hatley, C. von Riegen, and T. Rogers. Universal Description Discovery and Integration (UDDI) 3.0.2. Technical report, OASIS, October 2004.

- [CM+05] R. Chinnici, J.-J. Moreau, A. Ryman, and S. Weerawarana. Web Services Description Language. W3C, Version 2.0, Part 1, May 2005.
- [FBS04] X. Fu, T. Bultan, and J. Su. Analysis of interacting BPEL web services. In WWW '04: Proceedings of the 13th international conference on World Wide Web, pages 621-630. ACM Press, 2004.
- [Fer04] A. Ferrara. Web services: a process algebra approach. In Proceedings of ICSSOC, pages 242-251. ACM, 2004.
- [FFK04] Je. A. Fisteus, L. Sánchez Fernández, and C. Delgado Kloos. Formal Verification of BPEL4WS Business Collaborations. In Proceedings of the 5th International Conference on Electronic Commerce and Web Technologies (EC-Web '04), LNCS. Springer, August 2004.
- [FGV04] R. Farahbod, U. Glässer, and M. Vajihollahi. Specification and Validation of the Business Process Execution Language for Web Services. In: Abstract State Machines, volume 3052 of Lecture Notes in Computer Science, pages 78-94. Springer, 2004.
- [Fö05] D. Fötsch. Eine XML-basierte Transformationsmethode für modellgetriebene Web-Service-Architekturen: Vom ARIS-IE-Geschäftsprozessmodell zum BPEL4WS-Prozess. In: K.-P. Fähnrich, M. Thränert, P. Wetzel (Hrsg.): Umsetzung von kooperativen Geschäftsprozessen auf eine internetbasierte IT-Struktur. Leipziger Beiträge zur Informatik: Band III. ISBN 3-934178-52-9.
- [Ged06a] Gedilan GmbH. <http://www.gedilan.com>
- [Ged06b] Gedilan Technologies GmbH, Nautilus Benutzerhandbuch, Version 4.5, 2006
- [Got00] K. Gottschalk. Web Services Architecture Overview. IBM whitepaper, IBM developerWorks, 01 September 2000.
<http://ibm.com/developerWorks/web/library/w-ovr/>
- [Hol03] G. Holzmann. The SPIN Model Checker: Primer and Reference Manual. Pearson Education, 2003.
- [HSS05] S. Hinz, K. Schmidt, and C. Stahl. Transforming BPEL to Petri Nets. In: W.M.P. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, Proceedings of the Third International Conference on Business Process Management (BPM 2005), volume 3649 of Lecture Notes in Computer Science, Nancy, France, pages 220-235, September 2005. Springer-Verlag.
- [KBR04] N. Kavantzias, 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>.
- [LMS+06] N. Lohmann, P. Massuthe, C. Stahl and D. Weinberg. Analyzing Interacting BPEL Processes. BPM 2006.
- [Mar04] A. Martens. Verteilte Geschäftsprozesse - Modellierung und Verifikation mit Hilfe von Web Services. Dissertation, WiKu-Verlag Stuttgart, 2004.
- [MCK] Model-Checking Kit. Homepage.
<http://www.fmi.uni-stuttgart.de/szs/tools/mckit/>
- [MLZ06] J. Mendling, K.B. Lassen, U. Zdun: Transformation Strategies between Block-Oriented and Graph-Oriented Process Modelling Languages. In: F. Lehner, H. Nösekabel, P. Kleinschmidt, eds.: Multikonferenz Wirtschaftsinformatik 2006 (MKWI 2006), Band 2, XML4BPM Track, GITO-Verlag Berlin, 2006, ISBN 3-936771-62-6, pages 297-312

- [MS05] P. Massuthe, and K. 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.
- [MSS05] P. Massuthe, W. Reisig, and K. Schmidt. An Operating Guideline Approach to the SOA. *Annals of Mathematics, Computing & Teleinformatics*, 1(3):35-43, 2005.
- [NR02] M. Nüttgens, F.J. Rump: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In: PROMISE 2002, Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen, GI Lecture Notes in Informatics P-21, 64–77. Gesellschaft für Informatik, 2002.
- [OASISTC] OASIS Web Services Business Process Execution Language (WSBPEL) TC. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel
- [RSS05] W. Reisig, K. Schmidt, and Ch. Stahl. Kommunizierende Geschäftsprozesse modellieren und analysieren. Accepted for *Informatik - Forschung und Entwicklung*, 2005.
- [Sch05] K. Schmidt. Controllability of Open Workflow Nets. In: Jörg Desel and Ulrich Frank, editors, Enterprise Modelling and Information Systems Architectures, volume P-75 of Lecture Notes in Informatics (LNI), Bonn, pages 236-249, 2005. *Entwicklungsmethoden für Informationssysteme und deren Anwendung (EMISA, RWTH Aachen)*, Köllen Druck+Verlag GmbH.
- [T4B] Tools4BPEL. Homepage. <http://www2.informatik.hu-berlin.de/top/tools4bpel/>