

Peer-to-Peer-Technologie für unternehmensweites und organisationsübergreifendes Workflow-Management

Matthias Bender,^{*} Steffen Kraus,[†] Florian Kupsch,[‡] German Shegalov,^{*}
Gerhard Weikum,^{*} Dirk Werth,[‡] Christian Zimmer^{*}

1 Motivation

Workflow-Management ist eine reife Technologie; ihre Erfolgsbilanz beim möglichen Einsatz für die Steuerung unternehmensweiter und organisationsübergreifender Geschäftsprozesse ist aber eher bescheiden. Zwei neuere Technologien, die die Einsetzbarkeit von Workflow-Management vereinfachen und verbessern könnten, sind Web Services (WS) und Peer-to-Peer-Architekturen (P2P) [ACK04, Or01, Le03a]. Durch WS können Applikationen und ganze Sub-Workflows über Unternehmensgrenzen hinweg einheitlich gekapselt werden. P2P-Architekturen, für File-Sharing und Publish-Subscribe-Anwendungen sehr erfolgreich, könnten komplex strukturierte und organisatorisch schwierig zu vereinheitlichende Prozesse über lose gekoppelte Systeme dezentralisiert abwickeln, ohne in die Engpässe zentraler Prozessintegration und Systemadministration zu laufen.

Die Rolle von Web Services in Workflows ist Forschungsgegenstand in aktueller Literatur [Le03b], auch die Rolle von P2P-Technologie wurde bisher nur angedacht [vdAW01, SWSS03]. In dieser Arbeit wird eine Architektur vorgestellt, die Workflow-Management im Sinne einer P2P-Architektur dezentralisiert, ohne dabei Möglichkeiten zu Monitoring und Controlling einzubüßen. Mögliche Anwendungen für diesen Ansatz sind z.B. gegenseitige Leistungserbringungen verschiedener Unternehmen, bei denen Unternehmen einerseits Outsourcing an autonome Partner betreiben, andererseits aber jederzeit Eingriffs- und Überwachungsmöglichkeiten haben möchten. Ein konkretes Beispiel ist das Versenden von Produkten eines international tätigen Elektronikherstellers, der mit internationalen Frachtspeeditionen und einer Vielzahl nationaler Speditionen zusammenarbeitet und zur Kostenminimierung und Einhaltung von Lieferfristen Frachtwege dynamisch optimieren möchte.

^{*}Max-Planck-Institut für Informatik, Saarbrücken, {mbender, shegalov, weikum, czimmer}@mpi-sb.mpg.de

[†]DeepWeb GmbH, Saarbrücken, s.kraus@deepweb.de

[‡]Institut für Wirtschaftsinformatik, Saarbrücken, {kupsch, werth}@iwi.uni-sb.de

2 Problemcharakterisierung

Die IT-Landschaft vieler Unternehmen ist nach wie vor durch heterogene Insellösungen charakterisiert. Durch J2EE- oder Web-Service-Technologie gekapselte Anwendungssysteme können sich wechselseitig aufrufen, doch die eigentliche semantische Integration und die Steuerung globaler, langlebiger Geschäftsprozesse sind auf diese Weise nicht unterstützt. Einen Schritt weiter geht die EAI-Technologie (Enterprise Application Integration), bei der zusätzlich Datentransformationen zwischen den Methodenschnittstellen der Anwendungen spezifizierbar sind und ein Message-Broker zur Laufzeit die jeweils passende Anwendung aufruft und mit den richtigen Parametern versorgt [Ca04]. Workflow-Systeme andererseits setzen traditionell auf die Philosophie, dass ein Geschäftsprozess komplett spezifiziert werden muss. Einige akademische Forschungsprojekte haben interessante Ansätze zur verteilten Ausführung von Workflows entwickelt [RRD03, SGW01], doch hatten diese keinen Einfluss auf Produkte. Erst in jüngster Zeit werden zur Komposition von Prozessen über organisatorische Grenzen hinweg Web-Service-Erweiterungen wie BPEL4WS (Business Process Execution Language for Web Services) ernsthaft verfolgt [BPE]. Es ist jedoch unklar, ob und wie man damit autonom modellierte Prozesse integrieren kann und wie man die Workflow-Laufzeitumgebung mit Prozessüberwachungs- und -bewertungswerkzeugen organisationsübergreifend verbinden kann.

Wenn eine Fachabteilung eines großen Unternehmens einen neuen Geschäftsprozessstyp einrichten möchte und dafür eine Reihe bereits existierender Prozesse anderer Abteilungen als Subprozesse wiederverwenden möchte, stehen die IT-Abteilungen heute vor zwei gleichermaßen unbefriedigenden Alternativen:

- Die Fachabteilung identifiziert passende Subprozesse und bindet diese in den neuen Prozess mittels einfacher Message-Ports ein (z.B. mit SOAP-Nachrichten). Die Parameterbindungen zwischen Rufer und Gerufenem werden in EAI-Konnektoren festgelegt. Dies ist für jeden Interaktionspunkt zwischen Subprozessen einzeln zu spezifizieren, es gibt jedoch auf der Message-Ebene keinen expliziten Prozesskontext mehr, in dem der globale Fluss des Workflows festgehalten und überwacht wird. Dieser Ansatz verhält sich also lediglich wie ein Publish-Subscribe-System für zustandslose Dienste; zustandsbehaftete, langlebige Prozesse werden eigentlich gar nicht unterstützt, sondern auf der „mikroskopischen“ Ebene der ausgetauschten Nachrichten simuliert. Positiv an diesem Ansatz ist, dass er dezentralisiert arbeitet und im Prinzip keine globalen Engpässe beinhaltet. Jede Fachabteilung kann Message-Ports anbieten, jede kann ihre Prozesse an die Ports anderer Abteilungen binden. Dies entspricht der Philosophie von Peer-to-Peer-Systemen; allerdings gehen die globale Nachvollziehbarkeit und Qualitätssicherung auf der strategischen Ebene der Prozesse weitgehend verloren.
- Die für den neuen Prozess notwendigen Subprozesse werden identifiziert und in einem zentralen Repository explizit spezifiziert. Basierend auf dem Repository steuert eine Workflow-Engine die Ausführung der globalen Workflows. Dies entspricht dem „klassischen“ Integrationsgedanken, der sich in vielen Fällen als zu träge und aufwändig erwiesen hat. Der notwendige Aufwand für Aufbau und Pflege des glo-

balen Repositories wächst überproportional, und auch die Konfiguration und Administration einer unternehmensweiten Workflow-Engine mit den dahinter stehenden Applikations-Servern werfen signifikante Skalierbarkeits-, Beherrschbarkeits- und - wegen des hohen Personalaufwands für die Systemadministration - auch Kostenprobleme auf. Positiv an diesem Ansatz ist jedoch, dass alle Möglichkeiten zur Überwachung und Bewertung globaler Prozesse erhalten bleiben, sowohl hinsichtlich IT-orientierter Leistungsmerkmale wie z.B. Antwortzeiten als auch hinsichtlich betrieblicher Kenngrößen wie z.B. Durchlaufzeiten oder Terminüberschreitungen.

3 Die P2E2-Architektur

Im P2E2-Projekt [P2E] streben wir eine Architektur für organisationsübergreifende Workflows an, die die Nachteile der beiden oben ausgeführten Ansätze vermeidet, gleichzeitig aber deren Vorteile bewahrt und kombiniert: Prozesskopplung erfolgt dezentralisiert und dynamisch, die resultierenden globalen Prozesse sind jedoch nachvollziehbar und kontrollierbar, und die zugrundeliegende IT-Infrastruktur vermeidet Konfigurations- und Administrationsengpässe. Der Philosophie von P2P-Systemen folgend wird jede Organisationseinheit - jeder Peer - als autonom angesehen, die Kopplung zwischen den Workflow-System verschiedener Einheiten ist lose und hochgradig dynamisch, und es gibt im gesamten System keinerlei zentralisierte Komponenten. Die Dezentralisierung kommt dabei vor allem in der wesentlich leichteren, dynamischen Einrichtung und Konfiguration neuer Dienste und Prozesse zum Ausdruck. Ein ähnlich radikaler P2P-Ansatz wie File-Sharing-System kommt jedoch für Unternehmen wegen des Mangels an Qualitätssicherung nicht in Frage. P2E2 hebt sich von anderen P2P-Architekturen dadurch ab, dass es trotz Dezentralisierung Möglichkeiten zur Überwachung und Qualitätssicherung globaler Prozesse bietet.

Die charakteristischen Merkmale des P2E2-Ansatzes sind:

- Ein Geschäftsprozess kann von jedem Peer mit Hilfe eines beliebigen Design-Tools (z.B. Aris Toolset) spezifiziert werden. Ein Prozess kann Subprozesse enthalten, deren interne Abläufe explizit spezifiziert sind, zur Laufzeit aber an andere Peers delegiert werden. Umgekehrt können Aktivitäten, die aus Sicht eines Peers atomar sind, von einem anderen Peer ausgeführt werden, der die Aktivität mit Hilfe einer eigenen Prozessspezifikation in einen Subprozess auflöst.
- Jede Aktivität bzw. jeder Sub-Workflow erhält vom rufenden Peer einen Kontext, der mindestens die Eingabe- und Ausgabeparameter enthält, aber auch weitergehende Information über die bisherige Historie oder über die zur Fortsetzung nach Beendigung des Sub-Workflows aufzurufenden Folgeaktivitäten enthalten kann.
- Für die Interaktionen zwischen Peers sowie zur Verwaltung von Kontexten wird die Workflow-Engine bzw. der Applikations-Server jedes Peers durch einen P2E2-Adapter gekapselt. Dieser verbirgt mittels Web-Service- und J2EE-Technologie die Heterogenität unter den beteiligten Peers und verwaltet den P2P-Verbund im Sinne einer dynamischen Föderation.

- Zur Überwachung laufender Workflows und zum Zweck des betriebswirtschaftlichen Controllings abgeschlossener Prozesse melden alle Peers IT-technische Messdaten und ggf. betriebswirtschaftliche Kenngrößen an eine P2E2-Datenbank. Im gesamten P2P-System gibt es viele solcher Datenbanken; jeder Peer kann frei wählen, welche Daten in welcher Granularität er zu diesem Zweck an andere Datenbanken weitergibt und mit welchen Datenbanken er für diese Monitoring-Zwecke arbeiten will.
- Wenn ein Peer aktuelle Zustands- bzw. Controlling-Information über einen ihm bekannten Prozess ermitteln möchte, wendet er sich an einen Prozess-Performance-Manager (PPM), der seinerseits Daten aus einer P2E2-Datenbank extrahiert. Wenn die dort vorhandene Information unvollständig oder veraltet ist, wendet er sich seinerseits an andere ihm bekannte PPMs. Zusätzlich zu dieser Pull-Funktionalität läuft zwischen den verschiedenen P2E2-Datenbanken ein Push-basiertes Protokoll zur Informationsverbreitung im Sinne eines „Gossiping“-Protokolls.

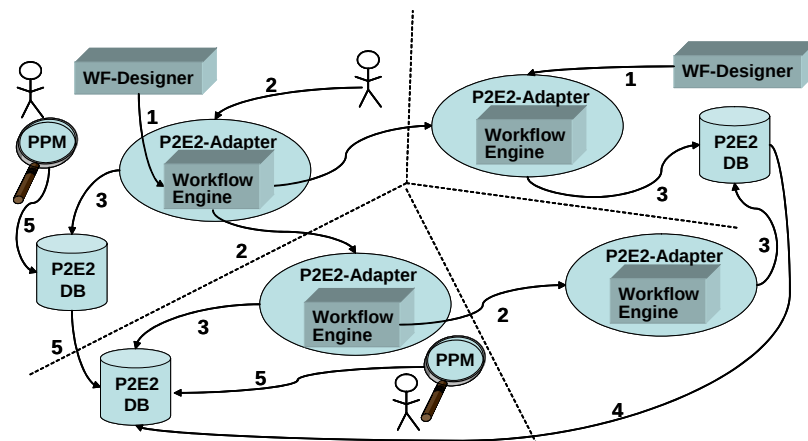


Abbildung 1: P2E2 Architecture

Hinsichtlich der Überwachung und des Controllings von Prozessen entspricht P2E2 einer flexiblen, dynamisch umkonfigurierbaren Publish-Subscribe-Architektur. Abb. 1 zeigt eine Ausprägung dieser Architekturprinzipien. Die gestrichelten Linien entsprechen organisatorischen Grenzen; es gibt also vier Peers in diesem Szenario. Die Pfeile geben den Steuerfluss zwischen den beteiligten Komponenten an, ihre Nummern entsprechen der Reihenfolge. In der Abbildung wird ein Workflow vom Peer links oben spezifiziert (1), und später wird eine Ausprägung dieses Prozesstyps gestartet (2). Eine der auf anderen Peers gestarteten Aktivitäten ist selbst ein gekapselter Sub-Workflow, der vom Peer rechts oben autonom spezifiziert wurde (1). Die verschiedenen Peers melden Ablaufdaten ihrer jeweiligen Workflow-Fragmente an P2E2-Datenbanken (3), und von dort wird ein Teil der Daten asynchron an andere P2E2-Datenbanken weitergemeldet (4). Wenn schließlich der den Workflow initiiierende Peer den Status seines Workflows abfragen will, z.B. weil er hochrechnen möchte, bis wann der Prozess voraussichtlich terminieren wird, wendet er sich an einen PPM, der eine oder mehrere P2E2-Datenbanken und andere ihm bekannte PPMs dazu befragt (5).

vor allem selbstorganisierenden IT-Infrastruktur für komplexe Geschäftsprozesse. Idealerweise sollte eine solche Infrastruktur sogar ein weitgehend automatisches „Matchmaking“ zwischen angebotenen Services und den für einen neu spezifizierten Prozess benötigten Subprozessen beinhalten [LR04, HBCS03]. Auf der technologischen Seite sollte sich ein solches P2P-System im Sinne von „Autonomic Computing“ selbst optimieren und dynamisch rekonfigurieren [IB, GWW02]. Offensichtlich ist es noch ein weiter Weg bis zu diesen Idealen. Im P2E2-Projekt planen wir durch neue Konfigurations- und Administrations-Werkzeuge für eine P2P-Umgebung dem Langzeitziel näher zu kommen und die Beherrschung komplexer IT-Systeme signifikant zu vereinfachen.

Literatur

- [ACK04] Alonso, G., Casati, F., und Kuno, H.: *Web Services - Concepts, Architectures and Applications*. Springer. Berlin;Heidelberg;New York. 2004.
- [Axi] Apache web services project (<http://ws.apache.org/axis/>).
- [BPE] Business process execution language for web services (<http://www.ibm.com/developerworks/library/ws-bpel/>).
- [Ca04] Carey, M. J.: Bea liquid data for weblog: Xml-based enterprise information integration. In: *ICDE*. S. 800–803. IEEE Computer Society. 2004.
- [GWW02] Gillmann, M., Weikum, G., und Wonner, W.: Workflow management with service quality guarantees (sigmod02). In: *SIGMOD Conference*. ACM. 2002.
- [Ha87] Harel, D.: Statecharts: A visual formalism for complex systems. *Science of Computer Programming*. 8(3):231–274. June 1987.
- [HBCS03] Hull, R., Benedikt, M., Christophides, V., und Su, J.: E-services: a look behind the curtain. S. 1–14. ACM. 2003.
- [IB] IBM. Autonomic computing (<http://www.research.ibm.com/autonomic/>).
- [Le03a] Lethin, R.: Introduction. *Commun. ACM*. 46(2):30–32. 2003.
- [Le03b] Leymann, F.: Web services: Distributed applications without limits. In: *BTW*. volume 26 of *LNI*. S. 2–23. GI. 2003.
- [LR04] Leymann, F. und Roller, D.: Modeling business processes with bpel4ws. In: *Modellierung*. volume 45 of *LNI*. S. 13–29. GI. 2004.
- [Or01] Oram, A.: *Peer-To-Peer; Harnessing the Benefits of a Disruption Technology*. O'Reilly. Beijing;Cambridge;Farnham. 2001.
- [P2E] Peer-to-peer enterprise environment (www.p2e2.de).
- [RRD03] Reichert, M., Rinderle, S., und Dadam, P.: Adept workflow management system. In: *Business Process Management*. volume 2678 of *LNCS*. Springer. 2003.
- [SGW01] Shegalov, G., Gillmann, M., und Weikum, G.: XML-enabled workflow management for e-services across heterogeneous platforms. *VLDB*. 10(1):91–103. 2001.
- [SMK⁺01] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., und Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: *SIGCOMM2001*. S. 149–160. ACM Press. 2001.
- [SWSS03] Schuler, C., Weber, R., Schuldt, H., und Schek, H.-J.: Peer-to-Peer Process Execution with OSIRIS. In: *Proceedings of ICSOC'2003*. S. 483–498. Trento, Italy. December 2003. Springer LNCS, Vol. 2910.
- [vdAW01] van der Aalst, W. M. P. und Weske, M.: The p2p approach to interorganizational workflows (caise2001). In: *CAiSE*. volume 2068 of *Lecture Notes in Computer Science*. Springer. 2001.