

# Prozessunterstützung für temporäre, ehrenamtliche und private Gruppen

Daniel Schulte

FernUniversität in Hagen, 58084 Hagen, Germany,  
Daniel.Schulte@FernUni-Hagen.de

**Zusammenfassung** Der Bedeutung von Geschäftsprozessen wird in der Dienstorientierung (*Service-oriented Computing*) durch Lösungen zur Automatisierung und Unterstützung bspw. mit WS-BPEL Rechnung getragen. Da allerdings ehrenamtlichen und privaten Gruppen weder entsprechendes Wissen noch notwendige Soft- und Hardwareprodukte zur Verfügung stehen, können sie derartige Prozessunterstützungen nicht für ihre Einsatzzwecke nutzen und setzen statt dessen weiterhin insbesondere auf E-Mails zur Koordination ihrer Prozesse. Diese Arbeit ermittelt daher die speziellen Anforderungen solcher Gruppen an eine Prozessunterstützung und stellt einen entsprechenden dienstorientierten Architekturstil vor.

## 1 Einführung

Computer und Internet unterstützen Anwender nicht nur bei Berechnungen und Informationsbeschaffung. Zusätzlich werden sie heute oft zur Koordination ganzer Prozesse in der Geschäftswelt aber auch in ehrenamtlich agierenden Gruppen wie z. B. Vereinen und im privaten Umfeld genutzt. Während die Steuerung von Prozessen in der Geschäftswelt bereits vielfältig adressiert und bspw. in dienstorientierten Architekturen mit WS-BPEL unterstützt wurde, sind bisherige Lösungen kaum für ehrenamtliche oder private Gruppen geeignet, da diese anderen Rahmenbedingungen unterliegen. Solche Gruppen nutzen bisher meist lokal installierte Software und E-Mails für ihre Prozesse. So werden initialer Einrichtungsaufwand, Einarbeitung und mögliche Kosten vermieden, aber Medienbrüche, unkontrollierte Informationsweitergabe, Verlust einzelner Informationen, mehrfache Datenhaltung, Dateninkonsistenz und Verlust der Prozessübersicht in Kauf genommen. Erfolgreiche Prozesse sind nur schwer wiederholbar, da Wissen über sie nur implizit vorhanden ist und somit ständig rekonstruiert werden muss. Kollaborative integrierte Prozessschritte wie bspw. das Sammeln von Tagesordnungspunkten sind nur durch erheblichen manuellen Aufwand realisierbar. Dabei machen vereinzelte Lösungen wie Konferenzmanagementsysteme deutlich, dass auch außerhalb des klassischen industriellen Umfeldes Prozessunterstützung hilfreich sein kann, um aufwendige und fehleranfällige händische Koordinationen zu vermeiden. Aufgrund unterschiedlicher Voraussetzungen sind aber Techniken aus dem Unternehmensumfeld nicht ohne weiteres für ehrenamtliche und private Gruppen adaptierbar.

Diese Arbeit analysiert daher in Abschnitt 2 die Anforderungen dieser Gruppen an eine Prozessunterstützung und leitet einen entsprechenden dienstorientierten Lösungsansatz her, der verschiedenen verteilten Diensten erlaubt, gezielt mit Anwendern in Kontakt zu treten. In Abschnitt 3 wird diese Lösung in Form eines Architekturstils präzisiert und in Abschnitt 4 eine ausstehende Fallstudie motiviert. Die Betrachtung verwandter Arbeiten in Abschnitt 5 und ein kurzer Ausblick in Abschnitt 6 beschließen diese Arbeit.

## 2 Prozesse für Gruppen

Prozesse von Gruppen, die nur vorübergehend existieren, deren Mitglieder nur temporär diesen Gruppen oder deren Mitglieder zugleich mehreren anderen unabhängigen Gruppen angehören (z. B. ehrenamtlich agierende Vereine, Veranstaltungen mit temporären Organisationsteams wie Benefizkonzerte, aber auch Workshops oder Konferenzen) haben gemein, dass sie primär aus manuellen Arbeitsschritten (Aufgaben genannt) bestehen, die von entsprechenden Mitarbeitern zu bearbeiten sind. Typische Rahmenbedingungen sind dabei:

- Es ist kein IT-Fachpersonal und oft nur private Hardware verfügbar.
- Es sind weder Zeit noch Geld für den initialen Aufbau einer technischen Infrastruktur verfügbar.
- Gruppen ändern sich beständig in ihrer Zusammensetzung oder arbeiten nur kurze Zeit zusammen.
- Gruppenbildung und Verantwortlichkeiten innerhalb dieser können auf Vereinstrukturen basieren, von Initiatoren bspw. im Rahmen von Workshops vorgegeben werden, oder auch z. B. bei Interessensgemeinschaften frei verhandelbar sein.
- Gruppenmitglieder arbeiten parallel an Projekten anderer Gruppen.
- Prozesse werden nur selten in einer Gruppe wiederholt, aber ähnliche Prozesse finden in vielen Gruppen statt (z. B. jährliche Mitgliederversammlungen).
- Prinzipielle Prozessabläufe sind kein Geschäftsgeheimnis.

Einige Forderungen an eine Prozessunterstützung sind demnach:

- Konzentration auf freie Technologien, Infrastrukturen, Dienste und Prozesse für weitestgehend kostenlose Lösungen
- Leicht formulier- und anpassbare Prozesse und Dienste; Ausnahmesituationen können beim Auftreten dynamisch gelöst werden
- Leichte Anwendung (und individuelle Anpassung) von Standardprozessen in verschiedenen Kontexten
- Einfacher und automatisierter Bezug von Aufgaben
- Unterstützung von Aufgaben verschiedener Art und Komplexität (von einfachen Überprüfungen bis hin zu komplexen, kooperativ zu bearbeitenden Aufgaben)
- Flexible Verwaltung von Gruppenmitgliedern und Verantwortlichkeiten

Insbesondere der einfache Zugang von Anwendern zu ihren Aufgaben soll in diesem ersten Schritt einer Lösungsentwicklung betrachtet werden. Denn — so weit vorhanden — setzen bisherige Lösungen wie Konferenzmanagementsysteme darauf, dass Anwender bestimmte Dienstseiten oder Portale regelmäßig aufsuchen, um sich über anstehende Aufgaben zu informieren. Dies weist aber zwei wesentliche Probleme auf: (1.) Der Anwender ist in einer Holschuld, er muss aktiv die entsprechenden Informationen über ausstehende Aufgaben einholen. Bei einer wachsenden Anzahl genutzter Dienste und einem zudem nur sporadischen Auftreten der Aufgaben ist eine zeitnahe Kenntnisnahme anstehender Aufgaben daher schwierig, so dass bisher oft ergänzend E-Mails eingesetzt werden. (2.) Die Verwendung von E-Mails kann zwar sicherstellen, dass Anwender zeitnah von Aufgaben erfahren, letztendlich werden aber weder die zeitliche Einplanung und Bearbeitung noch die Weiterverfolgung von Aufgaben nebst zugehörigen Prozessen unterstützt.

Die wichtigsten Anforderungen in diesem ersten Lösungsschritt lassen sich daher zunächst auf folgende Punkte reduzieren:

- Es darf — auch wenn Prozesse und Dienste mehrerer Anbieter (*Service Provider*) genutzt werden — nur einen Einstiegspunkt für den Anwender geben, an dem ihm alle wesentlichen Informationen zu ausstehenden Aufgaben und zur Verfolgung von Prozessen (ohne sein aktives Eingreifen) zur Verfügung gestellt werden.
- Dieser Einstiegspunkt muss einen flexiblen Zugriff auf zur Aufgabenausführung notwendige unterstützende Dienste und Informationen bieten.
- Dieser Einstiegspunkt muss von Prozessen und Diensten mit aktuellen Informationen, z. B. zum Status einer Aufgabe, versorgt werden können.

Die oft verwendeten E-Mails bieten durch ihre i. d. R. personenbezogene Adressierung zwar einen zentralen Einstiegspunkt, eignen sich aber nicht zur übersichtlichen Aufgabenverwaltung und -darstellung und erlauben zudem keine komfortablen Aktualisierungen bspw. des Aufgabenstatus. Die Verwendung von Portalen und proprietären Lösungen kann zwar eine gute Aufgabenverwaltung bieten, führt aber wieder zu dem Problem multipler Einstiegspunkte. Ein E-Mail-ähnlicher Ansatz mit strukturierten Informationen und einer dienstorientierten Schnittstelle kann beide Vorteile zusammenführen und personenbezogene Aufgabenzuweisungen aber auch Aktualisierungen von Informationen zu Aufgaben aus beliebigen Prozessen und Diensten heraus unterstützen.

Diese Idee wird im Folgenden in Form eines *Aufgabenlistendienstes* aufgegriffen, der über eine eindeutige, einem Anwender zugeordnete URI adressierbar ist und erlaubt, über im Detail noch zu entwickelnde Dienste Aufgaben inkl. Metainformationen wie Deadline, Status, o. ä. einem Anwender zuzuordnen. Der Anwender kann über diesen Dienst auf seine aktuellen Aufgaben zugreifen und diese verwalten, und sie bspw. auch nach Metadaten sortieren. Die Dienstorientierung erlaubt dabei sowohl auf Seiten des Dienstansbieters und des Prozesses als auch auf Seiten von Klientanwendungen für den Anwender flexible und innovative Lösungen unabhängig von verwendeten Programmiersprachen u. ä. Der nächste Abschnitt wird nun einen Architekturstil vorstellen, der dieser Idee entspricht.

### 3 Architekturstil

Nach Fielding [4] ist ein Architekturstil eine Menge von architektonischen Bedingungen, die die Rollen und Funktionen architektonischer Elemente und die erlaubter Beziehungen zwischen diesen für eine konforme Architektur einschränken. Ein Architekturstil für den Ansatz aus Abschnitt 2 wird in Abbildung 1 skizziert.

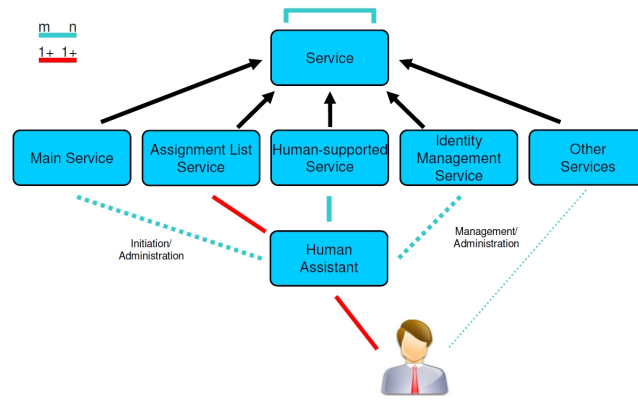


Abbildung 1. Architekturstil

Anwender werden durch eindeutige URIs, unter denen ein Aufgabenlistendienst (*assignment list service*) neu zugewiesene Aufgaben sowie Aufgabenaktualisierungen entgegennimmt, zu adressierbaren Entitäten (ähnlich einer E-Mail-Adresse nebst Postfach). Dienste, die den Eingriff von Anwendern benötigen (*human-supported services*), informieren diese per Aufgabenlistendienst über ausstehende Aufgaben und geben dabei neben einigen Metainformationen u. a. eine eindeutige URI für den Aufruf einer Repräsentation der Aufgabe an. Um größtmögliche Flexibilität bei der Aufgabenrealisierungen zu ermöglichen, werden Daten über potentielle Nutzer (z. B. bei der Dienst-Initialisierung übergeben) und interne Zustände von den Diensten eigenverantwortlich verwaltet. Eine Auslagerung dieser Funktionalitäten ist in Form weiterer Dienste oder Entwicklungsframeworks optional denkbar.

Zusätzliche Identitätsmanagementdienste (*identity management services*) erlauben eine späte, dynamische Bindung von Rollen an konkrete Anwender und eine Wiederverwendung von Organisationsstrukturen und Anwenderinformationen. Selbst Randbedingungen wie Urlaub oder Arbeitsauslastung ließen sich bei dieser dynamischen Bindung berücksichtigen. Den Diensten werden dann statt Adressen von Anwendern Rollenbezeichnungen und die Adresse eines Identitätsmanagementdienstes übergeben, sodass erst bei Auftreten einer Aufgabe die konkreten Anwenderadressen ermittelt und genutzt werden.

Lokale und netzbasierte Klienten (*human assistant*) erlauben den Anwendern den komfortablen Zugriff auf die sie betreffenden Aufgaben, und können bspw. durch Kalenderintegration zur Bearbeitungsplanung Mehrwerte bieten.

Für den praktischen Einsatz ist dieser Stil u. a. um standardisierte Schnittstellen für Identitätsmanagementdienste und Aufgabenlistendienste zu konkretisieren, um gemäß o.g. Anforderungen anbieterübergreifend arbeitende Lösungen zu ermöglichen. Über alle andere Dienste und Klienten sind dagegen möglichst wenig Annahmen zu machen, um flexible und innovative Realisierungen zu ermöglichen.

## 4 Fallbeispiel

Folgendes, noch umzusetzendes, einfaches Fallbeispiel ist zur Demonstration der Realisierbarkeit geplant: Für die Vorbereitung einer Sitzung, z. B. eines Mitarbeitertreffens eines Vereines oder einer öffentlichen Versammlung einer Bürgerbewegung, wird ein Prozess nebst notwendigen Diensten entwickelt werden, der das gemeinsame Sammeln von Tagesordnungspunkten und das Versenden von Einladungen zur Sitzung unter Angabe der Tagesordnung unterstützt. Der Prozess besteht aus folgenden Schritten:

1. Rahmendaten wie Datum, Raum, Sitzungsleitung, etc. sind zu spezifizieren (demonstriert die Initialisierung einer Prozessinstanz mit anwenderspezifizierten Daten inkl. Benennung von potentiellen Anwendern als Sitzungsleitung, etc. und die Konfiguration der Instanz z. B. bzgl. eines optionalen Anmelde-schrittes).
2. Tagesordnungspunkte sind zu sammeln (Beispiel für eine kooperative Aufgabe; wenn Anwender nicht gegenseitig ihre Beiträge zur Sammlung einsehen können, erhöht sich die Wahrscheinlichkeit von Mehrfachnennungen signifikant, einzelne Punkte können aber auch leichter vergessen werden).
3. Wenn der Sitzungsleitung nicht alle potentiellen Teilnehmer bekannt sind, ist ggf. eine Anmeldung z. B. über ein Webschnittstelle zu ermöglichen (Beispiel einer dynamisch wachsenden Gruppe).
4. Tagesordnung ist zu genehmigen/überprüfen (einfache Aufgabe).
5. Einladungen mit Tagesordnung sind an alle Teilnehmer zu versenden (klassischer, automatisierter Dienst).

Der Einsatz eines Identitätsmanagementdienstes ist optional zu ermöglichen, um das Potential der Wiederverwendung von Anwenderinformationen insbesondere für temporär stabile Gruppen aufzuzeigen. Gleichzeitig kann aber der Einsatz mit dem Beispiel einer Bürgerbewegung auch für dynamisch wachsende Gruppen demonstriert werden. Darüber hinaus wird die Fallstudie zeigen müssen, dass etablierte Lösungen wie z. B. YAWL [8] zur Prozesssteuerung adaptiert werden können, insbesondere solange der einfache Zugang von Anwendern zu ihren Aufgaben im Fokus steht.

## 5 Verwandte Arbeiten

Gängige Workflow- und Task-Management-Systeme bspw. von IBM, SAP oder Active Endpoints basieren u. a. auf WS-BPEL, BPEL4People und WS-HumanTask, um Prozesse in einem dienstorientierten Kontext zu modellieren und manuelle Aufgaben einzubinden. Alle drei Techniken entstammen dem Unternehmensumfeld und bieten daher für Gruppen im Fokus dieser Arbeit keine direkte Unterstützung: WS-HumanTask [1] spezifiziert Aufgaben (*Tasks*) als nicht-teilbare Arbeitseinheiten mit einem eigenen Lebenszyklus und einer programmiersprachenunabhängigen (aber XML-basierten) Schnittstelle insbesondere zur Verwaltung dieser Aufgaben. WS-BPEL [3] ist eine XML-basierte Sprache zur Orchestrierung von Diensten und wird durch BPEL4People [2] um eine Integration von Aufgaben (*Human Activities*) erweitert.

WS-BPEL und BPEL4People bieten interessante Möglichkeiten zur Formulierung von Prozessen. Im Rahmen einer allgemeinen und flexiblen Architektur ist aber von konkreten Sprachen soweit möglich zu abstrahieren, um auch alternative Implementierungen zu ermöglichen. WS-HumanTask dagegen ist zwar sprachunabhängig gehalten, unterstützt aber z. B. mit seinem Fokus auf nicht-teilbare Aufgaben keine kollaborativen Aufgaben (der definierte Lebenszyklus erlaubt keine derartigen Aufgabentypen). Die auf diesen Techniken basierenden Lösungen leiden für den hier anvisierten — aber bei deren Entwicklung nicht vorgesehenen — Einsatz an weiteren Problemen: Es gibt kein einheitliches Modell für Aufgabenlisten und -zuweisungen (Aufgabenverwaltung) und die Lösungen sind — da für das komplexe Unternehmensfeld mit hohen Anforderungen an Robustheit, Skalierbarkeit u. ä. entwickelt — für nicht professionelle Gruppen zu komplex und kostspielig. Der Einsatz eines Enterprise Service Bus bspw. ist nicht für das Internet und die Nutzung frei verfügbarer Dienste optimiert, die aber im Rahmen der o. g. Anforderungen hier wesentlich sind.

Das freie und leichtgewichtige Workflow-Management-System YAWL (Yet Another Workflow Language) [7, 8] ermöglichen den Einsatz einer Prozessunterstützung für verteilte Gruppen mit eingeschränkter IT-Infrastruktur wie bspw. von Ouyang et al. [5] beschrieben. Da jedoch Prozessausführung (in Form der *Workflow Engine*) und Aufgabenliste (*Worklist*) eng miteinander verzahnt sind, wird der Einsatz der integrierten Aufgabenverwaltung mit anderen Workflow-Management-Systemen nicht unterstützt. Das Ziel eines einzigen Einstiegspunktes für Anwender wird — sofern nicht alle Prozesse von einem Anbieter stammen — demnach nicht erreicht.

Selbst im geschäftlichen Umfeld führt der Mangel einer standardisierten Aufgabenverwaltung dazu, dass Anwender, die in mehreren Unternehmen oder Abteilungen mit eigener Softwareinfrastruktur involviert sind, mehrere Aufgabenlisten verwalten und abarbeiten müssen, und dass eine Auslagerung einzelner Aufgaben und Teilprozesse nicht unterstützt wird. Unger et al. [6] analysieren ausführlich Anforderungen an eine Lösung, die die unternehmensübergreifende Bearbeitung von Aufgaben ermöglicht, und stellen eine resultierende Architektur vor. Auch wenn einige gemeinsame Anforderungen wie z. B. die prinzipielle Forderung einer gemeinsamen Aufgabenverwaltung (*Task Management*) bestehen,

sind aufgrund der unterschiedlichen Rahmenbedingungen (bspw. (kein) Rückgriff auf IT-Fachpersonal und eigene IT-Infrastruktur) wesentliche Unterschiede festzuhalten, die in anderen Architekturstilen münden: Während bei Unger et al. [6] Task Engines für die Ausführung von einzelnen Aufgaben verantwortlich sind und Klienten auf mehrere Task Engines zugreifen können (den Klienten müssen also initial die Task Engines bekannt gemacht werden), wird in dieser Arbeit eine Anwenderadressierung eingeführt, die flexiblere Aufgabenzuweisungen und Aufgabenausführungsumgebungen ermöglicht, dafür aber bspw. die Einhaltung von Zuweisungsrichtlinien durchaus erschweren kann.

## 6 Zusammenfassung und Ausblick

Aufgaben von Anwendern in Prozessen können vielfältiger Natur sein: Einfache Genehmigungs- und Prüftätigkeiten oder kreative Aufgaben, alleine oder kooperativ auszuführende Aufgaben, der Organisation des Anwenders, einmaligen Veranstaltungen oder ehrenamtlich und privaten Projekten entstammende Aufgaben, und in bspw. Java, XML oder PHP realisierte Aufgaben. Letztendlich muss aber jede Aufgabe ihren Bearbeiter (Anwender) finden. Dazu wurde hier ein einfacher, von der E-Mail-Infrastruktur inspirierter Architekturstil vorgestellt, dessen Konkretisierung Gegenstand kommender Forschungsarbeit werden wird.

Auf diese Architektur aufbauend sind Datenschutz, Authentifizierung und Sicherung gegen Spam komfortabel zu ermöglichen. Aber auch die Flexibilität, Kontrollierbarkeit und Nachvollziehbarkeit ganzer Prozesse sind in Zukunft noch zu adressieren.

## Literatur

1. Agrawal, A. et al.: Web Services Human Task (WS-HumanTask), Version 1.0, 2007.
2. Agrawal, A. et al.: WS-BPEL Extension for People (BPEL4People), Version 1.0, 2007.
3. Alves, A. et al.: Web Services Business Process Execution Language Version 2.0, 2007.
4. Fielding, R.T.: Architectural Styles and the Design of Network-based Software Architectures. PhD thesis. University of California, 2000.
5. Ouyang, C., La Rosa, M.; ter Hofstede, A.H.M., Dumas, M. & Shortland, K.: Toward Web-Scale Workflows for Film Production. IEEE Internet Computing 12(5), 53 – 61, 2008.
6. Unger, T. & Bauer, T.: Towards a Standardized Task Management. Multikonferenz Wirtschaftsinformatik, GITO-Verlag, Berlin, 2008.
7. van der Aalst, W.M.P. & ter Hofstede, A.H.M.: YAWL: Yet Another Workflow Language. Information Systems, 30(4), 245 – 275, 2005.
8. YAWL Foundation: YAWL Yet Another Workflow Language. <http://www.yawl-system.com> [2009-02-20]