



Durchsichtiges Spiel

Human Tasks und ihre besondere Rolle für das Business Process Management

Hongguang Yang

In IT-Anwendungen für Unternehmen, in denen BPM (Business Process Management) eingesetzt wird, spielen die Benutzeraufgaben eine besondere Rolle. Die Systematik ihrer Behandlung ist nicht nur von komplexer Natur, sondern auch erfahrungsgemäß für die Akzeptanz des BPM-Ansatzes mitentscheidend. Im Unterschied zu den automatisierten Aktivitäten können sich die Benutzeraufgaben über eine wesentlich längere Zeit erstrecken und bezüglich ihrer Wechselwirkung mit dem zugrunde liegenden Prozessablauf vielfältiger sein. Benutzeraufgaben und ihre Standardisierungen mittels BPMN und WS-HumanTask sowie die unterschiedlichen Umsetzungen – anhand von zwei Beispielen aus dem Open Source (jBPM und Activiti) – bilden den Schwerpunkt dieses Artikels. Ebenso werden die Vor- und Nachteile der WS-HumanTask-Spezifikation für die Projektpraxis aufgezeigt, um den angehenden BPM-Anwendern eine Groborientierung zu ermöglichen.

Benutzeraufgaben für BPM

► In aktuellen BPM-Einsatzszenarien müssen die Prozessmodelle sowohl mit automatisierten Aktivitäten als auch mit Benutzeraufgaben (auch „Benutzeraktivitäten“, „User Tasks“ bzw. „Human Tasks“) adäquat umgehen können. Die automatisierten Aktivitäten werden durch die verfügbaren Servicekomponenten (beispielsweise Webservices) selbstständig ohne menschliches Zutun erledigt. Bei den Benutzeraufgaben handelt es sich um die Aktivitäten, die durch Menschen, normalerweise Sachbearbeiter im Kontext einer Organisationseinheit, erledigt werden müssen. In der Regel delegieren die Prozessmodelle ihre Benutzeraufgaben an eine Verwaltungskomponente (Human Task Manager), die zwischen der Ausführung der Aufgaben durch die zuständigen Sachbearbeiter und den Prozessmodellen „koordiniert“.

Wenn es in den Prozessmodellen gar keine Benutzeraufgaben gibt, so handelt es sich um automatisierte Prozesse, die im weiteren Sinne der „Batch-Verarbeitung“ entsprechen. Prozessmodelle können auch ausschließlich Benutzeraufgaben enthalten und alle Aktivitäten im Zusammenhang des Menschen abbilden, gänzlich ohne Automatisierung.

In der Projektpraxis führt der Begriff „Prozessautomatisierung“ häufig zum Missverständnis, denn er suggeriert die Automatisierung in Verbindung mit Prozessen. Der Prozessgedanke bzw. der BPM-Ansatz ist jedoch häufig mit unterschiedlichen (nicht automatisierbaren) Benutzeraufgaben verbunden. Das Ziel von BPM ist ja, die geschäftlichen Abläufe im Unternehmen bzw. ihre Unterstützung durch IT besser zu koordinieren, anstatt sie lediglich einseitig zu automatisieren. Ein anderes Problem in der Praxis ist, dass die undurchdringlichen Standardisierungen, zum Beispiel im Bereich von Workflow-Management, sowie unzureichende Werkzeuge zur Integration von Benutzeraufgaben vielen IT-Organisationen die Einführung von BPM erschwert.

Allmählich setzt sich mit BPMN 2.0 (Business Process Model and Notation) ein Standard durch [OMG11], der zum einen den IT-Organisationen die Erstellung von Prozessmodellen für die unterschiedlichsten geschäftlichen Abläufe mit oder ohne menschliche Interaktionen ermöglicht. Zum anderen, und dies fördert auch das Vordringen von BPMN 2.0 in die Praxis, stellen zahlreiche Open-Source-Frameworks und kommerzielle Produkte für diesen Standard die passenden Werkzeuge zur Modellierung (Process Designer) und zur Ausführung (Process Engine) bereit. Interessanterweise ist es diese Kombination von Modellierungswerkzeugen und Ausführungsmaschinen, die auch die Abstimmung zwischen Business (Fachabteilung) und IT (Entwicklung) effektiv fördert und verbessert [Freund12]. Das Vorgehen, dass die Fachabteilungen und die Anwendungsentwicklungen an denselben Prozessmodellen des Unternehmens mitwirken, setzt sich signifikant durch gegenüber den getrennten Modellhaltungen, die oft zu falschen Interpretationen der Abläufe führen.

Behandlung von Benutzeraufgaben in BPMN 2.0

In BPMN 2.0 werden unter der Kategorie „Human Interactions“ zwei Arten von Aktivitäten innerhalb des Prozessmodells definiert, die durch eine Beteiligung des Menschen als Benutzer erledigt werden sollen, nämlich „User Task“ und „Manual Task“ (s. Abb. 1).

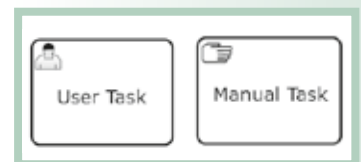


Abb. 1: Aktivitätsnotationen im Rahmen der Human Interactions in BPMN 2.0

Laut Definition in BPMN ist eine User Task ein typischer „Workflow“, der durch einen Task Manager eingeplant werden kann und den ein menschlicher Bearbeiter mit Hilfe einer Softwareanwendung abarbeitet. Ein Beispiel ist das Korrigieren der Adressdaten eines Kunden im CRM-System aufgrund einer Änderungsmitteilung.

Eine Manual Task ist eine Aktivität, die ein menschlicher Bearbeiter ohne Software-Hilfsmittel im Umfeld des Prozesskontextes erledigt, beispielsweise der lokale Anschluss eines Telefonapparats für einen Besteller durch einen Techniker.

Bei einer User Task interessiert das Prozessmodell, wann die Bearbeitung einer Aufgabe startet, eventuell wer sie an welche andere Person weitergibt und wann sie fertig ist. Möglicherweise kommt hier eine definierte Eskalation bzw. Weiterleitung ins Spiel, wenn die User Task innerhalb einer festgelegten Zeit nicht erledigt wird. Eine User Task hat einen Lebenszyklus, der durch den Task Manager im Kontext des Prozessmodells verwaltet wird. Hingegen stellt eine Manual Task eine „nicht-verwaltete“ Aktivität dar: Das Prozessmodell interessiert nur ihre bloße Existenz, aber kein weiteres Detail.

BPMN 2.0 weist auf die „WS-HumanTask“-Spezifikation (Web Services Human Task) von OASIS hin [OASIS10] und empfiehlt die Verwendung der dort definierten Attribute zur Beschreibung des Lebenszyklus einer User Task. BPMN 2.0 definiert auch einige rollenbezogene menschliche Bearbeiter („Human Performer“ bzw. „Potential Owner“ einer User Task) unter Nutzung der Terminologie aus WS-HumanTask.

Das sind dann auch schon die wesentlichen Kriterien der Human Tasks in BPMN 2.0. Außer der als verbindlich geltenden Empfehlung an WS-HumanTask ist die Umsetzung des Human Task Managers sowie seiner Interaktionen mit dem Prozessmodell aus Sicht von BPMN 2.0 fast regelfrei. Gerade

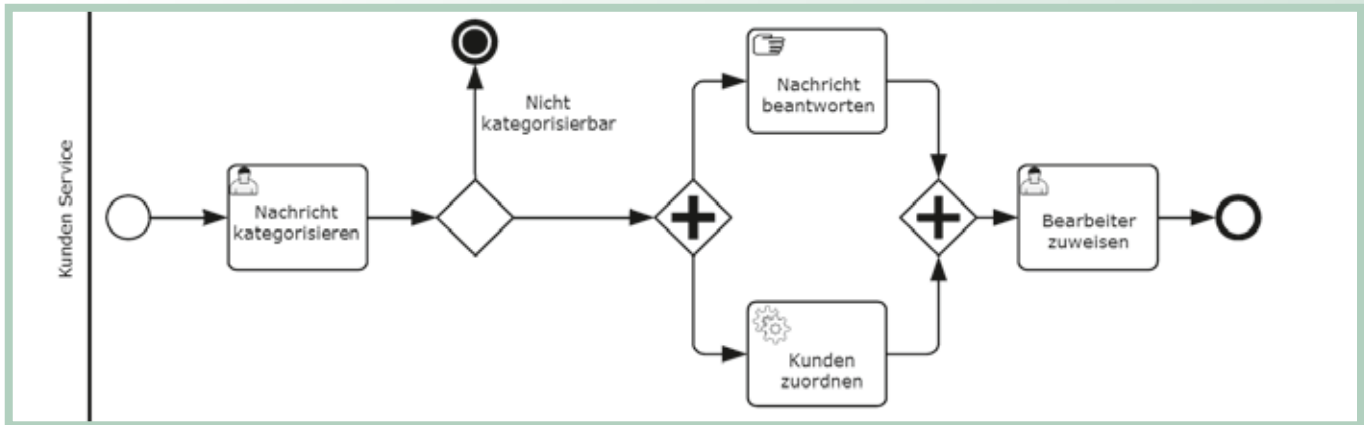


Abb. 2: Beispielprozess in BPMN 2.0-Notation

deswegen ist die Prozessnotation im Hinblick auf die menschliche Beteiligung auch einfach zu interpretieren.

Im Abbildung 2 ist beispielhaft ein Prozess in der BPMN 2.0-Notation anhand eines vereinfachten Modells für die (first-level) Bearbeitung von E-Mail-Anfragen durch die Kunden in einem Servicecenter wiedergegeben.

Der Prozessablauf kann in eine frühzeitige Terminierung münden, wenn die User Task „Nachricht kategorisieren“ kein Ergebnis liefert. Im Normalfall schreibt der Servicecenter-Mitarbeiter an den Kunden zurück („Nachricht beantworten“), was durch eine Manual Task modelliert wird: Auch wenn das Schreiben mithilfe einer Textverarbeitung bzw. eines E-Mail-

Systems erfolgt, interessiert hier aus Prozesssicht kein weiteres Detail mehr. Zur gleichen Zeit werden weitere Informationen über den Kunden – zum Beispiel Vertragspolice – automatisiert abgefragt („Kunden zuordnen“). Zum Schluss wird die Bearbeitung an die zuständige Abteilung weitergeleitet („Bearbeiter zuweisen“).

Behandlung von Benutzeraufgaben in WS-HumanTask

WS-HumanTask standardisiert den Lebenszyklus einer „Human Task“ – das ist der OASIS-Sprachgebrauch für die Benutzeraufgabe – in Form von Task-Status und den Status-Übergängen, unabhängig davon, ob es sich bei einer solchen Task inhaltlich um eine einfache GUI-Maske, um eine vom Benutzer gesteuerte Job-Verarbeitung oder aber um die langdauernde Bearbeitung eines komplexen Anwendungsfalls handelt. Die Komponentenarchitektur mit Kommunikationsschnittstellen wird ebenso festgelegt (s. Abb. 3).

Da die WS-HuamnTask für ein größeres Kontextumfeld als reines BPM erschaffen wurde, bedient sie sich der eigenen Begriffe für die Architekturkomponenten. „Task Parent“ entspricht dem Prozessmodell in BPM. „Task Processor“ ist der Human Task Manager und „Task Client“ und „Task UI“ repräsentieren die Schnittstellen zur Softwareanwendung, die der Sachbearbeiter zur Erledigung von Aufgaben verwendet. Auf die Erläuterung von weiteren Details in Abbildung 3, beispielsweise Webservice-Porttypdefinition (pt) und Operationen (op) für Schnittstellenkommunikation, wird hier verzichtet.

Die wichtigsten Status im Lebenslauf einer Human Task sind

- ▼ Created,
- ▼ Ready,
- ▼ Reserved,
- ▼ inProgress,
- ▼ Completed und
- ▼ Failed (im Fehlerfall).

Das Prozessmodell kann über den Besitz eines Status einer Human Task in Kenntnis gesetzt werden, indem die Kommunikationsschnittstelle automatisch bedient wird. Die Regeln und Umstände eines Übergangs (Transition) von einem zum ande-

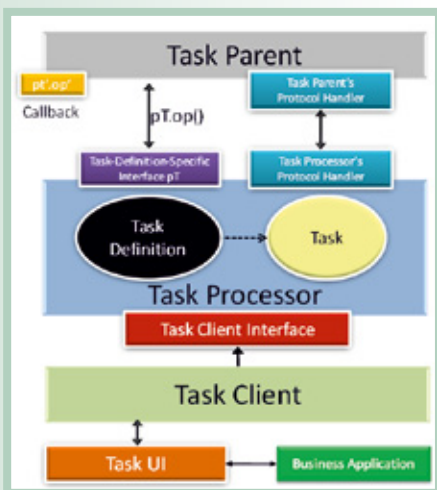


Abb. 3: WS-HumanTask Komponentenarchitektur mit Schnittstellen (Copyright © OASIS® 2011)

Status/Übergang	Created	Ready	Reserved	inProgress	Completed	Failed
Created	-	activation 1	activation 2	activation 3	-	-
Ready	-	forward	claim delegate	start	-	-
Reserved	-	release forward	delegate	start	-	-
inProgress	-	release forward	stop delegate	-	complete	failed
Completed	-	-	-	-	-	-
Failed	-	-	-	-	-	-

Tabelle 1: Wichtigste Status und Status-Übergänge einer Human Task



ren Status für eine Human Task sind im Detail festgelegt. In erster Linie lösen die Sachbearbeiter durch ihre Anwendungssoftware die Statusübergänge aus.

Die wichtigsten Statusübergänge werden in Tabelle 1 zusammengefasst. Beispielsweise geht eine neu erstellte Human Task (Status **Created**) durch die Methode **activation 3** in den Status **InProgress** (in Bearbeitung) über. Dieselbe Task mit dem Status **InProgress** wird dann durch die Methode **complete** in den Status **Completed** übergehen, was eine fehlerfreie Bearbeitung durch einen Benutzer kennzeichnet.

WS-HumanTask bietet damit ein sehr gutes Konzept zum allgemeinen Umgang mit Benutzeraufgaben. Diese sind für beliebige Projektsituationen anwendbar. Andererseits hat WS-HumanTask zwei sehr problematische Aspekte: Zum einen handelt es sich um ein allzu komplexes Schriftwerk, das für das eigentliche Thema (Benutzeraufgaben) eine recht hohe Einstiegshürde darstellt. Zum anderen schreibt das Werk bei einigen Subthemen ein zu rigides Vorgehen vor, beispielsweise manche Implementierungsdetails für die Webservices. Das hat dazu geführt, dass solche Vorgehen für die Praxis an Wert verlieren und die Softwarehersteller bei den Implementierungen dann doch ihren eigenen Weg am Standard vorbei gehen.

Dieses Abwägen von Nutzen und Aufwand spiegelt sich in den beiden Open-Source-Frameworks jBPM und Activiti wider, wie sie in den nächsten Abschnitten erläutert werden.

Handhabung von Benutzeraufgaben in jBPM

Das bekannte Java-Open-Source-Framework jBPM, aktuell in der Version 5.4 freigegeben [JBPM], unterstützt die Prozessmodellierung mit der BPMN 2.0-Notation, erlaubt die Simulation von Prozessabläufen für den Business-Designer und unterstützt die realen Abläufe durch eine Prozessmaschine (Prozess Engine). Für die Prozessmodellierung stehen gleich zwei Werkzeuge zur Verfügung, ein Eclipse-Plug-in und ein Browser-basierter Designer, der derzeit aufgrund der Art und Weise des Handlings mit dem Prozessmodell besser von den Business-Analysten bzw. von den Fachabteilungen angenommen wird.

Für die Prozessmodellierung und Prozessmaschine existieren eine Reihe von unterstützenden Softwarewerkzeugen, wie zum Beispiel das Repository-Werkzeug Guvnor für die versionierte Verwaltung der Prozessmodelle, sowie weitere Artefakte wie Prozesstestobjekte und Rules (s. Abb. 4). Die enge Integration von jBPM mit der Verarbeitung von Rules [Drools] – es handelt sich um dieselbe Ausführungsmaschine für die Prozesse und Rules – macht das Framework für die Anwendungsintegration in einem heterogenen IT-Umfeld noch interessanter.

In gewissem Sinne vererbt jBPM das Vorgehen von BPMN 2.0 hinsichtlich der Benutzeraufgaben konzeptionell weiter: Es stellt eine Implementierung von WS-HumanTask als „Standard-Implementierung“ bereit und erlaubt eine alterna-

tive Nutzung anderer projektspezifischer Implementierungen durch das Einbinden über die Java-Schnittstellen.

In der Standard-Implementierung berücksichtigt jBPM die WS-HumanTask-Spezifikation fast vollständig. Statt der Webservices stellt jBPM allerdings die Java-Methoden für den Task Client zur Verfügung (s. Listing 1). Damit ist die ursprüngliche Fokussierung von WS-HumanTask auf die Webservices bei dieser Implementierung nicht gegeben. Dafür ist die Nutzung in der Java-Laufzeitumgebung einfacher und bezüglich der wesentlichen Funktionalität in keiner Weise eingeschränkt.

```
public void start( long taskId, String userId, ... );
public void stop( long taskId, String userId, ... );
public void release( long taskId, String userId, ... );
public void suspend( long taskId, String userId, ... );
public void resume( long taskId, String userId, ... );
public void skip( long taskId, String userId, ... );
public void delegate( long taskId, String userId,
    String targetUserId, ... );
public void complete( long taskId, String userId,
    ContentData outputData, ... );
```

Listing 1: Java-Methoden für die Kommunikation eines Task Clients mit dem Task Manager

Die Einplanung von User Tasks erfolgt durch den Task Manager in einer eigenen Datenbank. Sowohl die Benutzer- und Gruppen-Zuweisung, die Weiterleitung als auch die Eskalation werden durch den Task Manager unterstützt. Die Einbindung in die Prozessmaschine manifestiert sich durch zwei unterschiedliche Komponenten zur asynchronen Kommunikation (Mina und HornetQ).

Die Nutzung von jBPM mit WS-HumanTask hat sich als sehr interessanter BPM-Ansatz zur Anwendungsintegration im heterogenen IT-Umfeld erwiesen, in dem die geschäftlichen Anforderungen an IT die Benutzeraufgaben intensiv erfordern [YangUhe12]. Abbildung 5 zeigt eine solche Integrationsarchitektur, die sich aus einem realen Projekt ergeben hat.

Dieser als „Best Practice“ herauskristallisierte Integrationsansatz zielt nicht auf die Vereinheitlichung von bestehenden Anwendungen, die in der Regel sehr unterschiedliche Benutzungsoberflächen und unterschiedliches Handling mit Daten vorweisen. Vielmehr manifestieren sich die vielfältigen (existierenden) Anwendungen als „Benutzeraufgaben“ in den Prozessmodellen. Dies erfolgt dadurch, dass die Anwendungen die WS-HumanTask-Schnittstellen und damit die dort definierten Status unterstützen. Diese Anwendungen können dann mit der Prozessmaschine interagieren. Die Verknüpfung der vorhandenen Business-Funktionalitäten aus verschiedenen Anwendungen zu den gewünschten geschäftlichen Abläufen

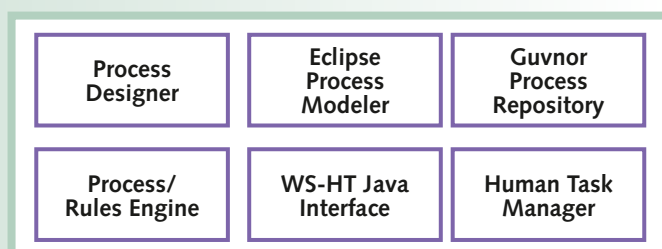


Abb. 4: jBPM-Toolset für BPM

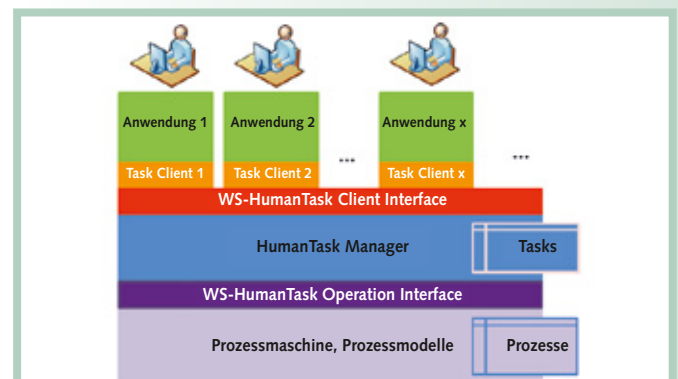


Abb. 5: Architektur zur Integration von Anwendungen mittels jBPM und WS-HumanTask

geschieht mittels WS-HumanTask und jBPM und unter minimal-invasivem Eingriff in die existierenden Anwendungen.

Handhabung von Benutzeraufgaben in Activiti

Das Java-Open-Source-BPM-Framework Activiti, aktuell in der Version 5.12 [Activiti], unterstützt ebenfalls die BPMN 2.0-Notationssprache durch ein Modellierungstool bzw. stellt die Prozessmaschine zur Ausführung bereit. Das Entwicklungsteam von Activiti verfolgt stärker als jBPM die Leichtgewichtigkeit ihres Frameworks. Das ist in der Softwarearchitektur soweit umgesetzt, dass das gesamte Framework als eine Bibliothek in einer beliebigen Java-Laufzeitumgebung eingebettet werden kann. Die wichtigsten Funktionalitäten sind in Form von „Services“ direkt in der Prozessmaschine gegeben, beispielsweise der „Repository Service“. Eine Installation und Anbindung von zusätzlichen Softwarekomponenten ist nicht erforderlich. Insbesondere gilt dieser Ansatz auch für unseren Betrachtungsfokus „User Tasks“.

Activiti bietet innerhalb der Prozessmaschine einen „Task Service“, der die „Human Interactions“ aus BPMN 2.0 ausprogrammiert und das Handling mit den Status und ihren Übergängen von Tasks unterstützt. Es gibt aber keine separate Softwarekomponente für den Task Manager. Im Gegensatz zu jBPM beansprucht Activiti nicht die funktionale Implementierung der WS-HumanTask-Spezifikation, nutzt jedoch inhaltlich die dortige Schnittstellendefinition auf pragmatische Weise. Insofern stellen die Schnittstellen für User Tasks eine direkte Umsetzung der entsprechenden Beschreibungsteile aus der BPMN 2.0-Spezifikation in Anlehnung an WS-HumanTask dar. Dennoch erweist sich die Java-Implementierung der User Tasks in Activiti als ausreichend für die praktische Anwendung.

Ein Problem bei dieser „Prozessmaschinen-nahen“ Implementierung des User-Task-Managements kann entstehen, wenn in einem Projektumfeld bereits ein anwendungsspezifischer Task Manager existiert und für die Prozessmodelle genutzt werden muss. Dann muss die Activiti-Implementierung des Task Managers durch die bereits existierende ausgetauscht werden. Alternativ können die beiden Task Manager zur Laufzeit miteinander synchronisiert werden. Auf jeden Fall sind Mehraufwände damit verbunden. In einem solchen Fall würde sich bei jBPM die lockere Kopplung zwischen der Prozessmaschine und dem User Task Manager als vorteilhaft erweisen, da ein Austausch des Letzteren sowohl im Konzept als auch in der Implementierung vorgesehen und leicht umzusetzen ist.

Dieser Artikel zielt nicht auf einen funktionalen Vergleich der beiden Open-Source-Frameworks jBPM und Activiti, sondern betrachtet die unterschiedlichen Ansätze zum Handling mit Human Tasks. Während bei jBPM eine komplette Softwarekomponente mit ihrer eigenen Datenbank für die User-Task-Verwaltung verwendet wird, sind bei Activiti die Task-Verwaltungsfunktionen in der Prozessmaschine selbst angesiedelt. Konzeptionell nutzen die beiden Implementierungen die WS-HumanTask-Spezifikation, wobei jBPM die dedizierte Softwarekomponente konsequenter umsetzt und dafür eine ausgezeichnete Softwareimplementierung bietet. Die von jBPM mitgelieferte Task-Verwaltungskomponente kann im Übrigen auch als Stand-alone-Anwendung zur Verwaltung von Human Tasks ohne Prozesskontext eingesetzt werden, was in vielen Projektsituationen die Anwendungsentwicklung entlasten kann, weil die Software für den Task Manager nicht mehr neu erstellt werden muss.

In der Summe kann man bei den beiden Open-Source-Implementierungen für die Benutzeraufgaben feststellen, dass sie die

„guten Kerne“ aus der WS-HumanTask-Spezifikation herausgepickt haben und jeweils auf ihre Art und Weise die rigiden, und eben auch nicht unbedingt nützlichen Dinge weggelassen haben.

Fazit

Die geschäftlichen Anforderungen von Unternehmen werden immer stärker mit den prozessorientierten Vorgängen in der IT verknüpft, die sich ihrerseits durch die koordinierte Einbindung von existierenden Softwareanwendungen und beteiligten Benutzern modellieren und ausführen lassen. Die Erstellung und Nutzung von Prozessmodellen bekommt die professionellen Unterstützungen durch BPMN, WS-HumanTask (in abgespeckten Varianten) und durch die Werkzeuge und Ausführungsmaschinen insbesondere aus den Open-Source-Implementierungen.

Der Begriff „Prozessautomatisierung“ kann diese Entwicklungen nicht adäquat widerspiegeln. Vielmehr stehen die Unternehmen vor der Herausforderung, diese Entwicklungen in ihrer Organisations- und Arbeitsstruktur so zu berücksichtigen, dass das gemeinsame Verständnis der Fachabteilungen und Anwendungsentwicklungen für die geschäftlichen Anforderungen sowie ihre Umsetzung in die IT-Praxis gefördert werden. Die IT-Abläufe können für das Management transparenter und nachvollziehbar werden.

An die Open-Source-Gemeinde und an die Standardisierungsgremien sei appelliert, dass für die Spezifikation der WS-HumanTask eine Überarbeitung in Richtung „essenzielle Verschlinkung“ stattfinden sollte, wenn sie sich wirklich als ein Standard im BPM-Umfeld etablieren will. Nach dem Motto: Weniger ist mehr.

Literatur und Links

[Activiti] <http://www.activiti.org/>

[Drools] <http://www.jboss.org/drools>

[Freund12] J. Freund, BPMN 2.0: Eine Zwischenbilanz, in: OBJEKTSPEKTRUM, 6/2012, s. a. http://www.sigs-datacom.de/fileadmin/user_upload/zeitschriften/os/2012/06/freund_05_06_12_n38x.pdf

[Guvnor] <http://www.jboss.org/guvnor>

[JBPM] jBPM Home Page, <http://www.jboss.org/jbpm>

[OASIS10] WS-HumanTask 1.1,

<http://docs.oasis-open.org/ns/bpel4people/ws-humantask/200803>

[OMG11] BPMN 2.0, <http://www.omg.org/spec/BPMN/2.0/>

[YangUhe12] H. Yang, Ch. Uhe, jBPM-Einsatz zur Modellierung und Ablaufsteuerung von Geschäftsprozessen mit Human Task Integration, in: JavaSPEKTRUM, 01/2012, s. a.

http://www.sigs-datacom.de/fileadmin/user_upload/zeitschriften/js/2012/01/yang_uhe_JS_01_12.pdf



Dr. Hongguang Yang beschäftigt sich seit über zehn Jahren mit Java-basierten Technologien und Anwendungen. Die Softwarearchitektur, Anwendungsentwicklung und Integration heterogener IT-Systeme für Unternehmen sind die Schwerpunktthemen seiner Arbeit. Er bedient Firmenkunden in verschiedenen Branchen, insbesondere im Bereich Business Consulting für Versicherungen. Es ist Mitarbeiter und Lead IT Consultant des IT-Unternehmens msg systems ag. E-Mail: hongguang.yang@msg-systems.com