

# Business Process Simulation

## selbst gemacht

Business Process Management (BPM) ist inzwischen zu einem Top Buzzword aufgestiegen. Korrekt verwendet, bergen prozessorientierte Architekturen große Potenziale. Jedoch ist sowohl das initiale Design als auch die Verbesserung von Geschäftsprozessen problematisch, da die resultierende Effizienz nicht vorausgesagt werden kann und Prozessfehler im Livebetrieb immense Kosten hervorrufen können. Die Simulation von Geschäftsprozessen soll genau hier helfen. Neben den Grundlagen wird auch ein Open-Source-BPS-Tool auf Basis von JBoss jBPM beschrieben.



von Bernd Rücker

**S**imulation ist eigentlich ein altbekanntes Thema in der Informatik. In vielen Bereichen wird Simulationstechnik bereits erfolgreich eingesetzt, oft wenn eine Entscheidung mit großen Investitionen verbunden ist. Ein viel zitiertes Beispiel ist der Bau eines Containerhafens. Im Bereich der Geschäftsprozesse ist Simulation noch relativ selten anzutreffen: Dies liegt vor allem daran, dass der Einsatz von so genannten Business Process Engines technisch noch nicht allzu ausgereift ist.

Business Process Engines können, kurz gesagt, Geschäftsprozessmodelle ausführen und erlauben somit die automatisierte Steuerung von Prozessen. Dies ermöglicht Prozessmodellierung nicht nur zum Zweck der Dokumentation, sondern auch zur Softwareentwicklung. Im Zusammenhang mit Simulation bedeutet dies, dass für BPS-Projekte keine speziellen und oft aufwändigen Simulationsmodelle geschrieben werden müssen, sondern das existierende Modell zum Einsatz kommen kann. Das

senkt den Aufwand erheblich und wird in Zukunft dazu führen, dass BPS eine wichtige Rolle zukommen wird.

Dieser Artikel beschreibt zuerst ein fachliches Anwendungsbeispiel, um dem Leser ein besseres Gefühl für BPS zu vermitteln. Dann werden Grundlagen zu BPM und Simulation gelegt und abschließend beschrieben, wie das Open-Source-Tool auf der Basis bestehender Komponenten entwickelt wurde. Ein im nächsten Heft folgender Artikel wird dann das Fachbeispiel exemplarisch mit

diesem Tool umsetzen. Weiterführende Informationen, sowohl zur Theorie als auch zur praktischen Umsetzung und zum Fachbeispiel, sind in [1] enthalten. Diese Master Thesis ist Grundlage des Artikels, das angesprochene BPS-Tool ist in diesem Rahmen entstanden.

### BPS und der Business Process Management Lifecycle

Die Idee eines Lifecycles (Lebenszyklus) ist nicht nur für alle möglichen Themen modern, sondern meist auch sinnvoll. Für das Gebiet des Geschäftsprozessmanagements gibt es viele verschiedene solcher Lifecycles, Abbildung 1 zeigt die Version aus [1]. Letztendlich ist die Grundidee einfach: Nachdem man Anforderungen analysiert hat, wird ein Prozess designed und dann implementiert bzw. um Implementierungsdetails angereichert. Ist dies geschehen, kann er auf die bereits angesprochene Engine deployed werden. Während dann Prozesse ablaufen, kann die Engine ständig aktuelle Daten über deren Zustand bereitstellen, was eine aktive Problemerkennung und Steuerung ermöglicht. Erkannte Probleme und Engpässe können somit analysiert werden und führen zu Prozessverbesserungen.

Ein wichtiges Thema ist dabei die Tool-Unterstützung. Wie in anderen Bereichen birgt der BPM Lifecycle besonders viel Potenzial, wenn er ohne Tool- bzw. Medienbrüche vonstattengeht oder zumindest alle Tools ineinandergreifen können. Werden Prozesse in einem Tool modelliert, sollten sie auch automatisch in eine ausführbare Prozesssprache wie XPDL oder BPEL übersetzt werden können.

Prozesssimulation sollte nun auch in dieser Tool-Kette aufgehängt sein. Dies würde es ermöglichen, „normale“ Prozessmodelle, um Attribute zur Simula-

tion angereichert, auch zur Simulation zu benutzen. Werden Prozesse nicht neu erstellt, sondern verbessert, kann diese Integration auch ermöglichen, historische Daten des Prozesscontrollings als statistische Eingangsdaten der Simulation zu verwenden.

### Simulationsziele

Wann kann Geschäftsprozesssimulation nun hilfreich sein? Typische Ziele sind [2]:

- Identifikation von Durchlaufzeit, Dauer oder Kosten eines neuen oder veränderten Prozesses
- Unterstützung der Kapazitäts- oder Einsatzplanung
- Vorhersage von Leistungsdaten bei veränderten Umgebungssituationen, beispielsweise der doppelten Anzahl der Anfragen zu Weihnachten
- Vergleich unterschiedlicher Prozessdesigns in Bezug auf bestimmte Kennwerte
- Vergleich unterschiedlicher Eingangsparameter

### Anwendungsbeispiel

Um die Idee der Prozesssimulation zu verdeutlichen, wird ein kleines Anwendungsbeispiel eingeführt. Der beispielhafte Prozess ist in Abbildung 2 dargestellt. Es handelt sich um die vereinfachte Abwicklung von Retouren bei einem Webshop. Der Shop prüft zurückkommende Ware in einem Schnelltest auf Funktionstüchtigkeit. Wird kein Fehler festgestellt, so wird die Ware von einem Techniker eingehend unter die Lupe genommen. Kann der Fehler nachvollzogen werden, bekommt der Kunde sein Geld zurück. Funktioniert die Ware auch nach der zweiten Prüfung problemlos,

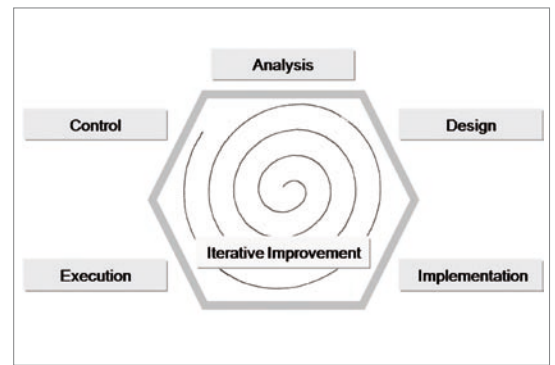


Abb. 1: BPM Lifecycle [1]

wird die Retoure abgelehnt und die Ware zurückgesendet.

Angenommen, der Prozess ist bereits automatisiert und läuft problemlos. Nun weiß man aber, dass nach der Weihnachtszeit besonders viel Ware zurückgesendet wird. Daher möchte man für diesen Prozess bei einer höheren Anzahl von Prozessinstanzen den Personalbedarf ermitteln. Klassisch würden bei diesem Problem die meisten von uns erst einmal zu einer Tabellenkalkulation greifen und etwas Vergleichbares mit Tabelle 1 aufstellen. Dazu werden statistische Daten wie Bearbeitungszeiten oder Wahrscheinlichkeiten an Verzweigungen aus dem Log der Business Process Engine abgefragt. Daraus kann man für jede Aufgabe die resultierende Anzahl von Aufgabeninstanzen berechnen. Da die Berechnung auf statistischen Daten basiert und damit eine gewisse Unsicherheit birgt, sollte man auch ein „Worst Case“-Szenario aufstellen. Aufgrund der Dauer im Normalfall und dem Worst Case können die Anzahl benötigter Personen sowie die dadurch entstehenden Kosten für beide Fälle ausgerechnet werden. Wie defensiv das Worst-Case-Szenario ausgeprägt sein

Aufgabe	Anzahl Instanzen	Dauer (normal/ Worst Case)	Anzahl benötigter Personen	Entstehende Kosten
	vorgegeben		berechnet	
transfer shipping costs	256	180 / 210	Buchhaltung: 1,71 / 1,99	1026 € / 1197 €
quick test	500	180 / 240	Sachbearbeitung: 3,13 / 4,17	1000 € / 1333 €
extended test	423	732 / 1180	Tester: 10,76 / 17,34	7745 € / 12483 €
Summe	500		Tester: 4,20 / 6,76 Sachbearbeitung...	11797 € / 17469 €

Tab. 1: Kalkulation von Ressourcenbedarf und Kosten als Tabelle

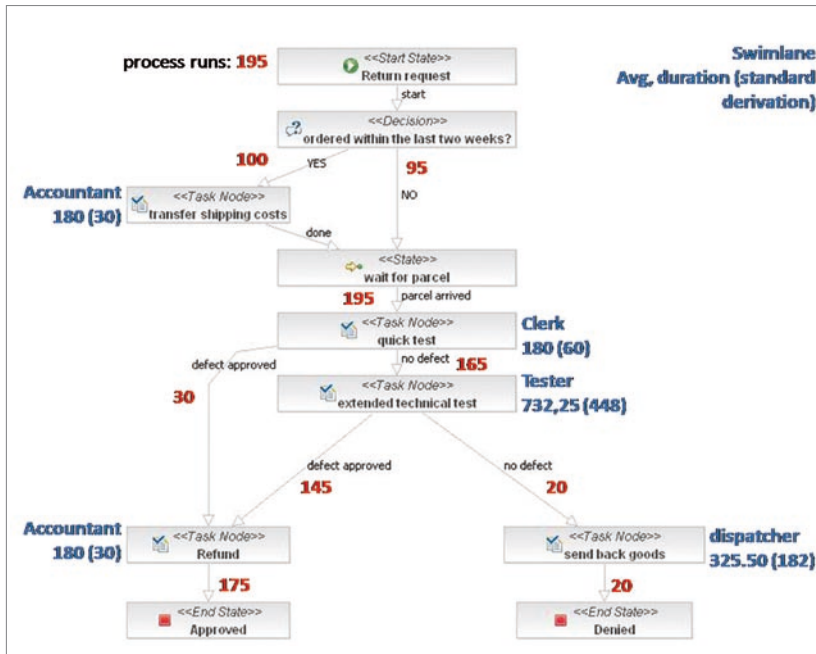


Abb. 2: Beispielprozess mit statistischen Kennzahlen [1]

soll, entscheidet der mögliche Schaden bei eventuellen Verzögerungen. Die Tabelle zeigt übrigens nur eine stark verdichtete Zusammenfassung, weitere Informationen können in [3] nachgeschlagen werden.

Dieses Vorgehen ist zwar besser als eine reine Bauchentscheidung, hat aber dennoch Nachteile. Zuallererst ignoriert die Tabelle jegliche statistische Theorie, was dazu führt, dass das Ergebnis immer noch eine grobe Schätzung darstellt und meist noch recht weit weg von der Wahrheit liegt. Oft fügt man wegen dieser Unschärfe unnötige Sicherheitszuschläge hinzu, die dann zu höheren Kosten führen. Des Weiteren werden solche Tabellen entsprechend kompliziert und unübersichtlich, wenn komplexere oder gar mehrere Prozesse beteiligt sind. Und oft sind es gerade die Querbeziehungen zwischen verschiedenen Prozessen, die zu Überraschungen führen können.

Für das beschriebene Problem kann man sich nun auch die Business Process

Simulation zu Nutze machen. Hier ist das Vorgehen ein anderes, da die Simulation die Unschärfe der Eingangsdaten bewusst berücksichtigt. Das Ergebnis ist aber noch keine Garantie für eine Personaleinsatzplanung. Die Simulation bietet aber den Vorteil, dass der präsentierte Simulationslauf problemlos wiederholt werden kann. Da in der Simulation die Stochastik am Werk ist, kommen für jeden Lauf unterschiedliche Werte heraus. Wie kann man davon nun auf ein fundiertes Ergebnis schließen, wenn jeder Lauf ein anderes Ergebnis hat? Der Trick ist, dass bei ausreichend häufiger Wiederholung die Ergebnisse gewissen Gesetzmäßigkeiten folgen und statistische Methoden ein Endergebnis sowie eine Fehlerwahrscheinlichkeit liefern. Aufgrund dieser Tatsache ist es nicht notwendig, Worst-Case-Szenarien zu definieren, das Simulationsergebnis liefert exakt, wie viel Ressourcen zu welcher Sicherheit führen.

Die Qualität steht dabei neben der

Wiederholungshäufigkeit auch mit der Qualität der Eingangsdaten in Verbindung. Werden diese aber wie beschrieben aus der laufenden Business Process Engine abgefragt und korrekt als Simulationseingangsdaten umgesetzt, so sollte die Qualität für die meisten Anwendungsfälle ausreichend sein.

Simuliert man das Beispiel aber trotzdem mit den gleichen Eingangsparametern, so erhält man das Ergebnis aus Tabelle 2. Zu den Ergebnissen seien zwei Dinge angemerkt: Erstens wurden in der Simulation für die Berechnung der „entstandenen Kosten“ nur beanspruchte Ressourcen mit Kosten belegt. Es wird also angenommen, dass nicht in Anspruch genommene Ressourcen nichts kosten. Dies entspricht zwar meist nicht der Realität, wurde aber in der Tabellenkalkulation auch angenommen (falls Sie dies nicht schon selbst gemerkt haben). Wird auch die gesamte Zeit mit Kosten belegt, erhielte man das Ergebnis in der rechten Spalte. Zweitens fällt auf, dass im Worst Case die Durchlaufzeiten besser sind als im Normalfall. Dies ist leicht zu erklären, da im Worst Case mehr Leute bereitgestellt werden, sodass Aufträge schneller abgearbeitet werden können. Und da nicht ausgelastete Ressourcen nicht zu Mehrkosten führen, sehen die Ergebnisse entsprechend rosiger aus. Hier sieht man bereits, dass man durch Simulation auch nicht sofort intuitive Zusammenhänge aufdecken kann.

### Optimierung

Was die Simulation nicht bewerkstelligen kann, ist Optimierung. Letztendlich prüft sie nur Hypothesen, die ihr als so genannte Szenarios präsentiert werden. Sie kann also nicht die Frage beantworten, welche Personaleinsatzplanung optimal ist, sondern lediglich vorgegebene Planung bewerten.

Die eigentliche Optimierung kann dann durch den Businessanalysten erfolgen, der das Tool bedient und sich an die richtige Lösung nach und nach „herantastet“. Dies wird allerdings mit steigender Komplexität problematisch, eine Alternative können Optimierungsalgorithmen darstellen. So könnte beispielsweise ein genetischer Algorithmus die optimale Personalbesetzung suchen und

Szenario	Auslastung Personen Ø	Prozess-durchlaufzeit Ø	Entstehende Kosten	Personalkosten gesamt
Weihnachten Normalfall	85,26 %	2.226 s	11.206 €	12.800 €
Weihnachten Worst Case	62,7 %	1.199 s	11.797 €	18.800 €

Tab. 2: Ergebnis der Simulation

dabei die Simulation zur Evaluierung der Kandidaten verwenden. Die Optimierung wird an dieser Stelle allerdings aus Platzgründen nicht weiter betrachtet.

### Grundlagen der Simulation

Nachdem schon einige Begriffe vorweg genommen wurden, soll hier noch ein kurzer Überblick der Simulationsgrundlagen gegeben werden. Simulation ist unter [4] folgendermaßen definiert: *Der Prozess, ein reales System zu beschreiben und dieses Model zum Experimentieren zu verwenden, um das Systemverhalten besser zu verstehen.*

Es gibt hauptsächlich zwei Arten der Simulation: Diskrete und kontinuierliche Simulation. Letztere wird beispielsweise für chemische Prozesse verwendet. Für BPS ist dagegen die so genannte *Discrete Event Simulation (DES)* interessant. Diese geht davon aus, dass ein Model durch eine endliche Anzahl an Zuständen gekennzeichnet ist und jede Zustandsänderung durch ein Event an einem definierten Zeitpunkt passiert. Dabei gibt es nur endlich viele Zeitpunkte und zwischen zwei benachbarten Zeitpunkten passiert nichts. Beispiele für solche Events sind das Ankommen zurückgesendeter Ware, der Abschluss der technischen Prüfung, die Abholung durch den Paketdienst oder Ähnliches.

Wichtig in DES-Systemen ist, dass die Modellzeit völlig unabhängig von der realen Zeit und damit auch der Dauer von Berechnungen oder der Geschwindigkeit des Simulationscomputers ist. Damit kann man sowohl berechnungsintensive Simulationen, die normalerweise in Millisekunden ablaufen, durchführen als auch zwei Jahre Einsatz eines Geschäftsprozesses in wenigen Minuten simulieren. Ein DES-System benötigt folgende Komponenten, um korrekt zu arbeiten:

- Ein Model, das immer den aktuellen Zustand hält
- Eine Simulationsuhr
- Eine Eventliste
- Einen Generator für Zufallszahlen
- Statistische Zähler und Datensammler
- Eine zentrale Steuerung, die für die Abarbeitung der Events zum richtigen Zeitpunkt zuständig ist

Im so genannten eventbasierten Modellierungsstil wird dabei jegliche Logik in Events untergebracht. Die Steuerung hält eine Liste der anstehenden Events nach Zeitpunkten sortiert und nimmt immer den ersten Event zur Hand. Dann stellt sie die Modellzeit auf den entsprechenden Zeitpunkt und bringt den Event in die Ausführung. Events können dabei durch die Simulations-Engine selbst oder extern erzeugt werden. In der unten beschriebenen Kombination mit einer Business Process Engine werden die meisten Events an definierten Stellen im Prozess erzeugt, zum Beispiel das „Aufgabe beendet“-Event im nächsten Absatz.

Letztendlich reichen diese einfachen Konzepte, um eine Simulationsumgebung aufzubauen. Mehr dazu und wie dies mit Java umgesetzt werden kann, können Sie unter [5] nachlesen.

### DES + BPM = BPS

Um ein DES-System für Prozesssimulation zu verwenden, kann es mit einer Business Process Engine verheiratet werden. Hier soll kurz das Prinzip erläutert werden. Automatisierte Geschäftsprozesse sind in einer Sprache wie BPEL oder, im Falle von jBPM, jPDL beschrieben. Zur Ausführungszeit kann die Engine Prozessinstanzen starten und steuern. Jede Instanz kann mit ihrer Umwelt in Verbindung treten, beispielsweise indem sie Services aufruft. Im Prozess gibt es Wartezustände, um auf asynchron eintreffende Ergebnisse oder externe Ereignisse zu warten. Und es gibt menschliche Interaktion, die normalerweise über eine Aufgabenverwaltung abgewickelt wird. Dabei kann der Prozess eine Aufgabe für eine Benutzergruppe anlegen, welche diese dann in ihrer To-Do-Liste sehen und bearbeiten kann.

In der Simulation kann der gleiche Prozess und die Engine auch verwendet werden, wobei einige Dinge zu beachten sind. Zuallererst ist in der Simulation keine menschliche Interaktion gewünscht. Dies kann dadurch gelöst werden, dass im BPS-Tool bei Anlage der Aufgabe ein Event erstellt wird, das die Beendigung der Aufgabe zu einer bestimmten Zeit in der Zukunft übernimmt. Die Bearbeitungszeit wird dabei als statistische Ver-

# das web von übermorgen.

webinale<sup>08</sup>  
the holistic web conference

26.–28. mai 2008  
kongresszentrum  
karlsruhe

## Speak-O-Rama



Glen Campbell  
Yahoo! Inc.



Duane Nickull  
Adobe Systems Inc.



Matthias  
Kannengiesser  
flashstar.de



Sascha Wolter  
Flashforum

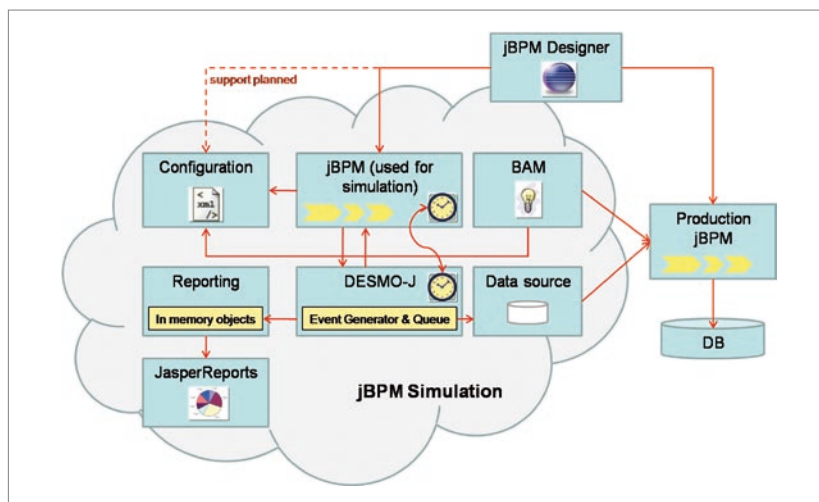


Abb. 3: Komponenten des Simulations-Tools

teilung benötigt und muss der Simulation bereitgestellt werden. Ähnlich verhält es sich mit Wartezuständen.

Serviceaufrufe sollten in der Simulation abgeschaltet, unverändert oder durch andere Serviceaufrufe ersetzt werden können. Denn meistens möchte man in der Simulation keine Änderungen an produktiven Systemen anstoßen, wie dies aber im produktiven Einsatz der Process Engine Alltag ist. Dies ist quasi eine Art Mocking für die Simulation.

Ebenfalls sind *Resource Pools*, die zur Verfügung stehende Ressourcen modellieren, wichtig. Wird ein Mensch oder eine Maschine zur Bearbeitung einer Aufgabe benötigt, so muss der Prozess diese Ressourcen anfordern. Stehen sie nicht zur Verfügung, muss die Prozessinstanz warten, bis sie frei werden. Hierfür bieten Simulationssysteme Queues an.

Abgesehen von vielen Detailproblemen, die es noch zu lösen gibt, reicht eine solche modifizierte Business Process Engine in Verbindung mit einem Discrete-Event-Simulation-System zur Abbildung der Prozesssimulation.

### Statistik

Simulation hat zwangsläufig viel mit Statistik und Stochastik zu tun. Statistische Verteilungen und Zufallszahlen spielen eine große Rolle. Zu Zufallszahlen sei an dieser Stelle nur gesagt, dass es im Computer ohne spezielle Hardware nur so genannte Pseudozufallszahlen gibt, welche von bestimmten Algorithmen ausgehend,

von einem Startwert, dem so genannten *Seed*, berechnet werden. Dadurch ergibt sich der Vorteil, dass die Zufallsdatenströme reproduzierbar sind, wenn man den gleichen Samen verwendet.

Wozu werden aber nun statistische Verteilungen benötigt? Angenommen, man hat historische Daten der Bearbeitungszeit zur Hand, so sind diese immer in ihrer Zahl begrenzt und bestimmte Werte werden nie vorkommen. Eine theoretische Verteilung gießt die Werte in eine Formel, die dann prinzipiell den gesamten möglichen Wertebereich abdeckt, allerdings bestimmte Zahlen mit einer Wahrscheinlichkeit produziert, die denen der historischen Probe entspricht. Eine solche Verteilung zu finden, ist nicht immer trivial, allerdings gibt es zahlreiche Statistikprogramme auf dem Markt, die einen bei diesem Vorhaben unterstützen. Beispielhaft sei das freie Tool GSTAT2 [6] genannt.

Die Verteilungsfunktionen, die meistens nur ein oder zwei Parameter benötigen, werden als Input der Simulation zur Verfügung gestellt, und zwar für Bearbeitungs- und Wartezeiten sowie für die Zeit zwischen zwei Prozessstartevents, also beispielsweise das Stellen eines Retourenantrags im Web.

Die Statistik hält weitere interessante Aspekte bereit, die im Rahmen der Simulation Anwendung finden. Ein Beispiel ist die Erkennung von so genannten „warm-up“-Zeiten, in denen die Simulation noch keine gültigen Werte liefert, oft weil Queues noch unrealistischer-

weise leer sind. Ein anderes Beispiel ist die Berechnung der Anzahl von Simulationsläufen, die benötigt wird, um eine bestimmte Genauigkeit beim Ergebnis hinzubekommen. Für diese weiterführenden Themen sei auf [1], [5] oder [7] verwiesen.

### Der Markt

Der Markt für BPM-Tools ist heute leider extrem unübersichtlich und verändert sich schnell. Zu überblicken, welche Tools BPS unterstützen, ist alleine schon schwierig, ganz gemein wird es, da mit „Simulation“ von verschiedenen Herstellern oft unterschiedliche Dinge gemeint sind. Neben der hier beschriebenen stochastischen Simulation ist auch oft die Animation von einzelnen Prozessdurchläufen gemeint. Auch wenn dies ein interessantes Feature eines BPM-Tools ist, es erlaubt auch oft ein Debugging von Prozessinstanzen, so ist es kein BPS im eigentlichen Sinne.

Eine Evaluierung im Vorfeld der Thesis hat gezeigt, dass zwar die großen Hersteller in ihren BPM-Suiten nach und nach BPS zur Verfügung stellen oder schon unterstützen, jedoch waren selten konkrete Detailinformationen zu finden. Auch gab es kein einziges BPS-Tool, welches Open Source zur Verfügung steht. Da dies allerdings BPS für eine breitere Masse an Benutzern zugänglich macht und auch zukünftige Forschung erleichtert, war genau die Erstellung eines solchen Tools Inhalt der angesprochenen Master Thesis.

### Das entwickelte BPS-Tool

Für die Entwicklung des Tools wurde auf bestehende Open-Source-Komponenten zurückgegriffen. Dabei kam JBoss jBPM [8] als Process Engine und DESMO-J der Uni Hamburg [9] als Simulations-Framework zum Einsatz. Das entwickelte Tool wurde im jBPM-Projekt committed und steht entsprechend unter LGPL zur Verfügung.

Das Zusammenspiel der Komponenten ist in Abbildung 3 illustriert und in [1] genauer beschrieben. Herzstück ist die Verbindung von jBPM und DESMO-J, wobei die Process Engine so erweitert wurde, dass sie die oben beschriebenen Events automatisch anlegt

und an DESMO-J übergibt. jBPM arbeitet in der Simulation ohne Persistenz, da ein Simulationslauf immer komplett im Speicher abläuft und somit keine Datenbank benötigt wird. DESMO-J arbeitet lediglich die Events ab und übernimmt die Steuerung der Modellzeit, die auch mit der in jBPM vorhandenen Uhr synchronisiert wird. In der Eventausführung werden dann beispielsweise Prozesse gestartet sowie Aufgaben oder Wartezustände beendet.

DESMO-J zeichnet ständig Kennzahlen wie die Wartezeit auf Ressourcen oder Prozessdurchlaufzeiten auf. Die Ergebnisse werden nach dem Simulationslauf als Java-Objekte zur Verfügung gestellt und können entweder automatisch ausgewertet oder auch mit einem Reportgenerator grafisch angezeigt werden. Im entwickelten Tool kommt hierbei JasperReports [10] zum Einsatz.

#### Ausblick

Der Artikel hat Grundlagen zu Business Process Simulation gelegt. Des Weiteren hat er die mögliche Kombination einer bestehenden Business Process Engine

mit einem bestehenden Simulations-Framework skizziert und gezeigt, dass dies auch mit Open-Source-Komponenten möglich ist.

Da BPM ein immer wichtigeres Thema wird und die kontinuierliche Verbesserung von Geschäftsprozessen nicht erst seit KAIZEN oder Six Sigma Bedeutung hat, wird auch Business Process Simulation in Zukunft einen höheren Stellenwert einnehmen. Das vorliegende Open-Source-Tool ermöglicht jedem den Einsatz von BPS, zumindest als erste Gehversuche. Die Forschung wird an Themen wie genetischer Verbesserung von Geschäftsprozessen dran bleiben und es ist abzuwarten, wann BPM-Suiten selbstständig Prozessverbesserungen vorschlagen. Somit lohnt es sich, das Konzept genauer unter die Lupe zu nehmen.

Ein anschließender Artikel im nächsten *Java Magazin* zeigt dann die konkrete Umsetzung des eingeführten Fachbeispiels mit dem Simulations-Tool. Wer also Sourcecode und Konfigurationsdetails vermisst hat, sei auf die nächste Ausgabe vertröstet. ■



**Bernd Rücker** ist Berater, Coach und Geschäftsführer bei der camunda GmbH. Sein besonderes Interesse liegt dabei im Bereich BPM und SOA sowie deren praktische Umsetzung, gerne auch mit Open Source. Er hat berufsbegleitend sein Studium als Master of Science in Software Technology erfolgreich abgeschlossen, in diesem Rahmen ist die angesprochene Master Thesis entstanden. Kontakt: [bernd.ruecker@camunda.com](mailto:bernd.ruecker@camunda.com).

#### Links & Literatur

- [1] Bernd Rücker: Master Thesis: [www.camunda.com/content/publikationen/bernd-ruecker-business-process-simulation-with-jbpm.pdf](http://www.camunda.com/content/publikationen/bernd-ruecker-business-process-simulation-with-jbpm.pdf)
- [2] Jörg Becker, Martin Kugeler, Michael Rosemann: Prozessmanagement. Ein Leitfaden zur prozessorientierten Organisationsgestaltung. Springer, 2007
- [3] jBPM Simulation Tutorial: [www.camunda.com/jbpm\\_simulation/jbpm\\_simulation\\_tutorial.html](http://www.camunda.com/jbpm_simulation/jbpm_simulation_tutorial.html)
- [4] Robert E. Shannon: Systems Simulation: The Art and Science. Prentice-Hall, 1975
- [5] Bernd Page and Wolfgang Kreutzer: The Java Simulation Handbook. Simulating Discrete Event Systems with UML and Java. Shaker, 2005
- [6] GSTAT2: [www.statoek.wiso.uni-goettingen.de/user/fred/gstat2.htm](http://www.statoek.wiso.uni-goettingen.de/user/fred/gstat2.htm)
- [7] Manuel Laguna, Johan Marklund: Business Process Modeling, Simulation, and Design. Prentice Hall, 2004
- [8] JBoss jBPM: [labs.jboss.com/jbossjbpm/](http://labs.jboss.com/jbossjbpm/)
- [9] DESMO-J: [www.desmoj.de/](http://www.desmoj.de/)
- [10] JasperReports: [www.jasperforge.org/jaspersoft/opensource/business\\_intelligence/jasperreports/](http://www.jasperforge.org/jaspersoft/opensource/business_intelligence/jasperreports/)

# das web von übermorgen.

**webinale**<sup>08</sup>  
the holistic web conference

**26.–28. mai 2008**  
kongresszentrum  
karlsruhe

## Speak-O-Rama



**Paul Ortchanian**  
Blast Radius



**Oliver Scheer**  
Microsoft  
Deutschland



**Jens Grochtdreis**  
SinnerSchrader



**André Michelle**  
Hobnox