

„EVENT-DRIVEN BUSINESS PROCESS MANAGEMENT“: EINE NEUE TECHNOLOGIE UND ERSTE PROJEKTE AM BEISPIEL DER DHL

2006 wurde „Event-Driven Business Process Management“ (EDBPM) als Begriff für eine neue Technologie-Plattform eingeführt. Dabei handelt es sich um eine Kombination von zwei verschiedenen Disziplinen: „Business Process Management“ (BPM) und „Complex Event Processing“ (CEP). Die internationale Organisation der Event Processing Technical Society (EPTS) kümmert sich um die Sammlung von Anwendungsfällen, um Standards, eine Nomenklatur und ein Glossar für Begriffsdefinitionen. Dieser Artikel versucht den aktuellen Stand der Kunst darzustellen. Am Beispiel eines Projekts für die Deutsche Post/DHL wird der Einsatz ausführlich erklärt. Weitere Projektbeispiele, z. B. aus dem Bankenbereich, zeigen, dass diese Technologie in vielen Anwendungsdomänen zum Einsatz kommen wird.

Im März 2006 wurde *Complex Event Processing (CEP)* offiziell beim 1. CEP-Symposium in Hawthorn/NY als Disziplin gegründet (vgl. [CEP06]). Im September 2008 wurde anlässlich der 3. EPTS-Konferenz in Stamford/CT die Organisation *Event Processing Technical Society (EPTS)* ins Leben gerufen (vgl. [EPTS07]). Neben anderen Schwerpunkten wird in diesem Zusammenhang seit 2006 unter dem Begriff *Event-Driven Business Process Management (EDBPM)* von der EPTS ein neuer Begriff für eine neue Technologie-Plattform eingeführt.

Dabei handelt es sich um eine Kombination von zwei unterschiedlichen Disziplinen: *Business Process Management (BPM)* und *Complex Event Processing (CEP)*. BPM basiert auf der Annahme, dass die Art und Weise, wie ein Unternehmen seine Geschäftsprozesse betreibt, seine Einzigartigkeit ausmacht. Aus diesem Grund werden Geschäftsprozesse heute als das wertvollste Gut eines Unternehmens betrachtet, das unter ganz unterschiedlichen Gesichtspunkten diskutiert werden kann. In diesem Artikel beschränken wir uns mehr auf die technischen Aspekte.

Unter BPM wird im Folgenden eine Softwareplattform verstanden, mit deren Hilfe ein Unternehmen Geschäftsprozesse modellieren, verwalten und optimieren

kann, sodass es seine Ziele bestmöglich erreicht. Zu diesem Zweck analysiert und verarbeitet ein parallel laufendes CEP-System – vom BPM-System unabhängig – Ereignisse (*Events*), die im Kontext zu den Geschäftsprozessen stehen bzw. von diesen ausgelöst werden. BPM- und CEP-System korrespondieren mittels Ereignissen, die von einem BPM-System und den IT-Services, die an die einzelnen Geschäftsprozess-Schritte geknüpft sind, erzeugt werden. Das CEP-System soll die Wertschöpfungskette und dementsprechend die einzelnen Geschäftsprozesse in Echtzeit überwachen. Auf Basis der Korrelation von einzelnen, atomaren Ereignissen zu komplexen Ereignissen können sich anbahnende Probleme frühzeitig erkannt und proaktive Maßnahmen oder Frühwarnungen veranlasst werden. In diesem Beitrag werden erste Projekte und Studien zu diesem Thema vorgestellt.

Historischer Hintergrund

Der Begriff EDBPM wurde erstmals 2003 in einem *White Paper* der Bruce Silver Associates im Zusammenhang mit der FileNet-P8-BPM-Plattform benutzt (vgl. [Bru03]). Dort wurde EDBPM noch als Synthese von Workflow-Management-System und *Enterprise Application Integration (EAI)* verstanden. Daneben wurde



Dr. Rainer von Ammon

(E-Mail: rainer.ammon@citt-online.com)

ist Initiator des internationalen Studiengangs EDBPM und Leiter des Centrum für Informations-Technologie Transfer (CITT) GmbH, das ein internationales Netzwerk zwischen Industrie, Technologie-Herstellern, Anwendern und Hochschulen bereit stellt.



Christoph Emmersberger

(E-Mail:

christoph.emmersberger@senacor.com)

und



Florian Springer

(E-Mail: florian.springer@senacor.com)

sind als Business IT Consultants bei Senacor Technologies tätig. Sie begleiten IT-Transformationsprojekte mit dem Schwerpunkt serviceorientierte Architekturen bei namhaften Kunden in der Automotive- und Logistikbranche. Ihre Erfahrung im Bereich EDBPM haben sie sowohl während des Studiums als auch in ihrer Beratertätigkeit aufgebaut.

bereits ein Konzept für Ereignisverarbeitung und Echtzeit-Business Activity Monitoring (BAM) beschrieben, allerdings nur als einfache Verarbeitung von Einzelereignissen ohne jeglichen Bezug zu CEP, da zu diesem Zeitpunkt CEP noch nicht als eigene Disziplin eingerichtet war. Obwohl das Buch „The Power of Events“ von David Luckham (vgl. [Luc02]) bereits 2002 publiziert wurde, war es bis zu dessen Vorstellung auf einer Konferenz des Markt-

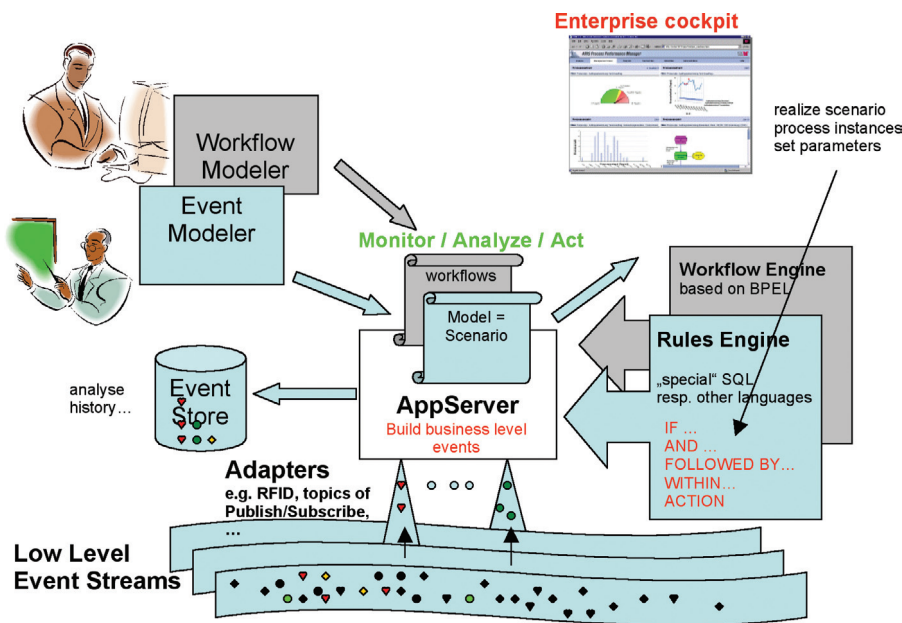


Abb. 1: Referenzmodell für das prinzipielle Zusammenwirken der wesentlichen CEP/BAM/BPM-Komponenten (aus [Amm07]).

forschungsunternehmens Gartner 2004 in Orlando kaum bekannt und es existierten bis dahin auch noch keine wirklich einsatzfähigen CEP-Plattformen. Erst im November 2007 wurde in einem Interview mit Ruma Sanyal von BEA Systems der Begriff EDBPM explizit als eine Kombination aus BPM und CEP genannt (vgl. [Dan07]).

Das Prinzip EDBPM und ein Referenzmodell

Abbildung 1 zeigt, nach welchem Prinzip BPM- und BAM/CEP-Plattformen auf der Basis von Ereignissen miteinander interagieren. Die grau dargestellten Komponenten kennzeichnen die Geschäftsprozessmodellierung und das Geschäftsprozessmanagement, die blau dargestellten Komponenten zeigen die wichtigsten Bestandteile eines Echtzeit-BAM auf der Basis von CEP. Seit einigen Monaten sind erste industriell einsetzbare CEP-Plattformen am Markt verfügbar, die bereits zu einem integrierten EDBPM vom selben Hersteller angeboten werden (z. B. von BEA-Systems, demnächst auch von IBM und Oracle) oder die mit anderen BPM/ BAM-Produkten verschiedener Hersteller kombiniert werden können (z. B. von Apama, Coral8, Esper, Systar). Wie erste Erfahrungen zeigen und auch schon in [Sch06]

prognostiziert wurde, benötigen solche Projekte in Zukunft mit dem Ereignis-Modellierer (*Event-Modeler*) einen weiteren Spezialisten, der wegen ganz unterschiedlicher Anforderungen nicht in Personalunion mit dem derzeitigen Business-Analysten bzw. Geschäftsprozess-Modellierer zu finden ist und erst an Hochschulen ausgebildet werden muss.

Der Geschäftsprozessmodellierer

Der Geschäftsprozessmodellierer identifiziert – ausgehend von der Wertschöpfungskette eines Unternehmens – Geschäftsprozesse, die er analysiert, reorganisiert und optimiert. Dabei verwendet er ein Modellierungswerkzeug, dem bisher im deutschsprachigen Raum häufig die proprietäre ARIS-Notation *eEPK* (*erweiterte Ereignis-gesteuerte Prozesskette*) zu Grunde lag, die aber in Zukunft durch eine standardisierte Notation ersetzt wird. Das Rennen hat im Winter 2007/2008 die *Business Process Modeling Notation* (BPMN) gewonnen, die 2002 von IBM definiert, später von der OMG standardisiert und im Februar 2006 in der Version 1.0 veröffentlicht wurde (vgl. [OMG06]). Ebenfalls im Winter 2007/2008 gab die IDS Scheer AG bekannt, auch BPMN zu unter-

stützen und den eigenen langjährigen Quasistandard der *eEPK* zu ersetzen.

Ein Geschäftsprozessmodell ist ein Netzwerkgraph, der auf Symbolen für Flussobjekte basiert, wie z. B. Aktivitäten, Gateways und Ereignisse. In Abhängigkeit von ihrer Position in einem Netzwerk kategorisiert BPMN die Ereignisse als Start-, Zwischen- und Stop-Ereignisse. Jede Ereigniskategorie hat verschiedene Ereignistypen, wie z. B. zeitliche Ereignisse, Nachrichten-Ereignisse und Ausnahme-Ereignisse. Die aktuelle BPMN-Version 1.1 kennt bereits 25 Ereignistypen, eine neue, vor allem auch um zusätzliche Ereignistypen überarbeitete, Version BPMN 2.0 ist für das dritte Quartal 2008 angekündigt. Die Modellierung von Geschäftsprozessen ist eine sehr anspruchsvolle Aufgabe für Geschäftsprozessmodellierer bzw. Business-Analysten, aber völlig verschieden von der Aufgabe eines Ereignis-Modellierers für echtzeitfähige, auf CEP basierend BAM-Anwendungen. Der Ereignis-Modellierer arbeitet mit verschiedenen Ereignistypen, die von Geschäftsprozess-Instanzen oder anderen Ereignisquellen erzeugt werden, wie z. B. SNMP-Traps, RFID-Nachrichten, Logfile-Einträge und JMS Publish/Subscribe-Topics.

Die Geschäftsprozessmodelle können in eine ausführbare Notation für eine *Workflow Engine* als Bestandteil einer BPM-Plattform transformiert werden. Ausführbare Notationen werden gegenwärtig noch standardisiert, etwa von OASIS als *Business Process Execution Language* (BPEL) oder von der Workflow Management Coalition als *XML Process Definition Language* (XPD). Wenn ein Geschäftsprozessmodell tatsächlich ausführbar sein soll, muss dieses nochmals fein-granularer remodelliert werden und zwar direkt in der BPM-Plattform und nicht im BP-Modellierungswerkzeug. Aus Sicht der IT-Abteilung müssen z. B. so genannte *Compensations* und *Exceptions* mit modelliert oder ergänzt werden, falls ein IT-Service nicht ausgeführt werden kann oder scheitert. Diese Aufgabe kann nicht von einem Business-Analysten erledigt werden, der nichts von den Interna eines IT-Services weiß, der mit einem Geschäftsprozess-Schritt assoziiert ist. Deshalb wird bereits heute mit den gerade auf den Markt kommenden BPM-Plattformen und den neuen Standards eine



neue Vorgehensweise für die Modellierung und Implementierung von Geschäftsprozessen benötigt. Business-Analysten aus den Fachabteilungen müssen eng mit IT-Spezialisten zusammenarbeiten, indem sie zusammen den Prozess und alle seine assoziierten IT-Services in gemeinsamen Sitzungen modellieren. Der Prozess wird dabei als BPMN visualisiert, aber bereits direkt in einem integrierten Modellierungswerkzeug einer BPM-Plattform, die auch eine Benutzungsschnittstelle für das Einpflegen der technischen Informationen enthält (wie z. B. „porttypes“, „ports“, „partnerlinks“ und „peoplelinks“). Diese Ergänzungen werden für die Prozessausführung z. B. durch eine BPEL-Engine benötigt. Die derzeit entstehenden BPM-Plattformen können außerdem automatisch Ereignisse generieren, wenn ein Service aufgerufen wird, nicht ausgeführt werden kann, scheitert oder ähnliches. Das alles sind Ereignistypen, die der Ereignis-Modellierer kennen muss, um BAM-Dashboards für einzelne Prozesse oder unternehmensweite *Enterprise Cockpits* realisieren zu können.

Der Ereignis-Modellierer

In Kooperation mit Prozesseignern der Fachabteilungen oder sogar mit der obersten Führungsebene (C-Level-Management) eines Unternehmens muss der Ereignis-Modellierer definieren, welche BAM-Sichten auf welche Weise in einem Dashboard angezeigt werden sollen, welche Warnungen an welche Gruppen im Unternehmen zu senden sind und welche Aktionen automatisch gestartet werden sollen, falls ein bestimmtes Ereignismuster erkannt wird. Ausgehend von solchen BAM-Sichten sucht der Ereignis-Modellierer nach den erforderlichen Ereignistypen und ihren Instanzen in den so genannten Ereignisströmen eines Unternehmens oder in *Event Stores*, wo diese als „historische Ereignisse“ gespeichert sind. Es ist eine hoch qualifizierte Aufgabe, die richtigen Ereignismuster für eine echtzeitfähige BAM-Sicht zu bestimmen. Der Ereignis-Modellierer muss nicht nur die verschiedenen Ereignisquellen beherrschen, sondern außerdem die entsprechenden Ereignis-Adapter installieren, die von der CEP-Plattform entweder vorgefertigt zur Verfügung gestellt werden (z. B. beinhaltet das Produkt von Syster mehr als 90 vorge-

fertigte Adapter) oder die der Ereignis-Modellierer erst noch entwickeln muss.

Programmiersprachen für Ereignisverarbeitung

Der Ereignis-Modellierer definiert die Ereignismuster für eine BAM-Sicht mithilfe einer *Event Processing Language (EPL)*. Gegenwärtig gibt es noch keinen Standard für eine EPL. Die Mitglieder der erwähnten CEP-Community, die als eigene Disziplin beim ersten CEP-Symposium gegründet wurde, diskutieren bislang die Frage nach einem geeigneten Standard für eine EPL sehr kontrovers. Es scheint, dass es verschiedene EPL-Ansätze für unterschiedliche Anwendungsdomänen geben wird, z. B. für *Algorithmic Trading* und für BAM-Anwendungen in der Logistik-, Finanz- oder Automobil-Domäne. Gegenwärtig sammelt die frisch gegründete *Event Processing Technical Society (EPTS)* (vgl. [EPTS07]) Use-Cases und klassifiziert diese nach unterschiedlichen Kriterien, wie z. B. Domänen-Zugehörigkeit (vgl. [CEP08]). Die CEP-Plattformen verwenden entweder eine SQL-ähnliche Sprache (z. B. Coral8, Esper, Oracle, StreamBase), verfolgen einen regelbasierten EPL-Ansatz (z. B. AMiT von IBM) und/oder sie bieten eine Benutzungsschnittstelle, die eine wie auch immer gartete Programmiersprache verdeckt und Code generiert (z. B. Tibco, AptSoft).

Die Modelle der Geschäftsprozesse und die Ereignisszenarien werden auf eine Middleware verteilt, z. B. einen Applikationsserver, der dann für die Hochverfügbarkeit, Skalierbarkeit und Transparenz verantwortlich ist.

Spezielle EDBPM-Anforderungen an die Datenbanktechnologie

In Verbindung mit EDBPM gibt es aus der Sicht der Datenbanktechnologie vor allem zwei größere Herausforderungen:

Aus *BPM-Sicht* besteht die Herausforderung darin, dass Millionen von Instanzen lang laufender Geschäftsprozesse und deren Status gespeichert werden müssen. Ein großes Unternehmen wie eine Bank oder ein Automobil-Hersteller hat mehrere tausend Geschäftsprozesse, z. B. verschiedene Arten von Kontoeröffnungen, Prozesse für Kreditanträge, Produktions-

und Verkaufsprozesse. Jeder Geschäftsprozess hat wiederum mehrere tausend Prozessinstanzen, z. B. einen Kreditantrag. Im Fall von lang laufenden Geschäftsprozessen muss die BPM-Plattform den Status all dieser Prozessinstanzen speichern, z. B. wenn auf die Unterschrift des Kreditantragstellers gewartet werden muss. Da dies Tage oder Wochen dauern kann, müssen innerhalb eines bestimmten Zeitfensters gleichzeitig Millionen von aktiven Geschäftsprozessen vorgehalten werden. Daraus können in der Praxis erhebliche technische Probleme, wie *Overflows of Queues* und Performanceprobleme, entstehen. Des Weiteren bereitet auch die Menge der Daten erhebliche Schwierigkeiten und Datenbanken müssen auf die Verarbeitung dieser hohen Anzahl hin optimiert werden.

Aus *BAM/CEP-Sicht* besteht die Herausforderung in der Verarbeitung von tausenden von Ereignissen, die pro Sekunde durch Geschäftsprozesse, deren assoziierte IT-Services sowie andere Ereignisquellen erzeugt werden. Für so genannte *In-Memory*-Datenbank-Technologien, mit denen Ereignisse in bestimmten Zeitfenstern für eine Echtzeitverarbeitung vorgehalten werden können, gibt es diverse Patente, z. B. „StreamBase“ von der RTM GmbH. Dabei ist eine im Arbeitsspeicher agierende SQL-Engine so konzipiert, dass Datenbankabfragen ausgeführt werden können, ohne dass Daten bzw. Ereignisse auf der Festplatte gespeichert werden müssen. Eine derartige SQL-Engine ermöglicht auch das Speichern von Ereignissen als „historische Ereignisse“ in einem *Event Store* für eine beliebige spätere Verarbeitung. Diese können bei Bedarf mit den aktuellen Ereignissen aggregiert und korreliert werden. Die grundlegende Idee besteht darin, die Ereignisse durch eine Vielzahl verschiedener *In-Memory*-Filter zu leiten, sozusagen als gespeicherte SQL-Anweisungen, und zu beobachten, welche Ergebnisse erzielt werden, anstatt eine Anzahl von SQL-Anweisungen gegen auf Festplatte gespeicherte Daten auszuführen. Das ist eine der Grundvoraussetzungen für Echtzeit-BAM und die Idee des *Predictive Business* (vgl. [Ran06]).

Seit Kurzem sind die ersten EDBPM-Lösungen auf dem Markt verfügbar. Derzeit wird versucht, für Use-Cases aus vertikalen Domänen Referenzmodelle zu erstellen und

diese als Prototypen zu implementieren sowie in *Proof of Concepts (PoC)* in den verschiedenen Domänen zu erproben, z. B. im Bereich Logistik, für das Finanzwesen, im Telekommunikationsbereich oder im Automobilssektor. Im Frühjahr 2008 wurde eine Studie für den Logistikbereich abgeschlossen (vgl. [Emm08]).

EDBPM im Bereich Logistik

Im Rahmen der Studie wurde ein typischer Geschäftsprozess der Deutschen Post, der die Use-Cases „Paketversand“, „Paketverfolgung“ und „Beschwerdemanagement“ umfasst, modelliert und implementiert. Die Geschäftslogik wurde auf Basis des *Enterprise Service Bus (ESB)* und des *SOA-Frameworks* der Post implementiert. Dieses Framework ist mittlerweile unter dem Namen *SOPERA* als Open-Source-Lösung verfügbar und bietet neben Standard-ESB-Funktionalität auch die Möglichkeit, Prozess- und Serviceinformationen als Ereignisse an die *CEP- und BAM-Komponenten* zu senden. Für den PoC wurden die verschiedenen Oracle-Komponenten in den ESB integriert:

- Oracle *BPEL (Business Process Management)*,
- Oracle *BAM (Business Activity Monitoring)* und
- eine Vorläuferversion von Oracle *CEP (Complex Event Processing)*.

Folgendes Beispiel zeigt vereinfacht den untersuchten Use-Case:

Ein Paket wird mittels verschiedener Transportmittel (z. B. Flugzeug, Lkw, Lieferwagen) auf einem festgelegten Weg durch unterschiedliche Regionen transportiert. In jeder Region können Ereignisse auftreten, die eine pünktliche Auslieferung des Pakets verhindern. So verhindert z. B. ein Stau auf der Autobahn das Vorankommen des Lkw oder ein Sturm den Start eines Transportflugzeugs. **Abbildung 2** zeigt einen vereinfachten Ausschnitt aus dem Paketzustellungsprozess. Darin wird einem Paket ein Transportmittel zugewiesen, mit dem es die Strecke zurücklegt, bis es beim Empfänger angekommen ist. Während der Auslieferung nehmen verschiedene Ereignisse Einfluss auf den Lieferprozess. In diesem Beispiel sind das Verkehrs-, Wetter- und RFID-Ereignisse.

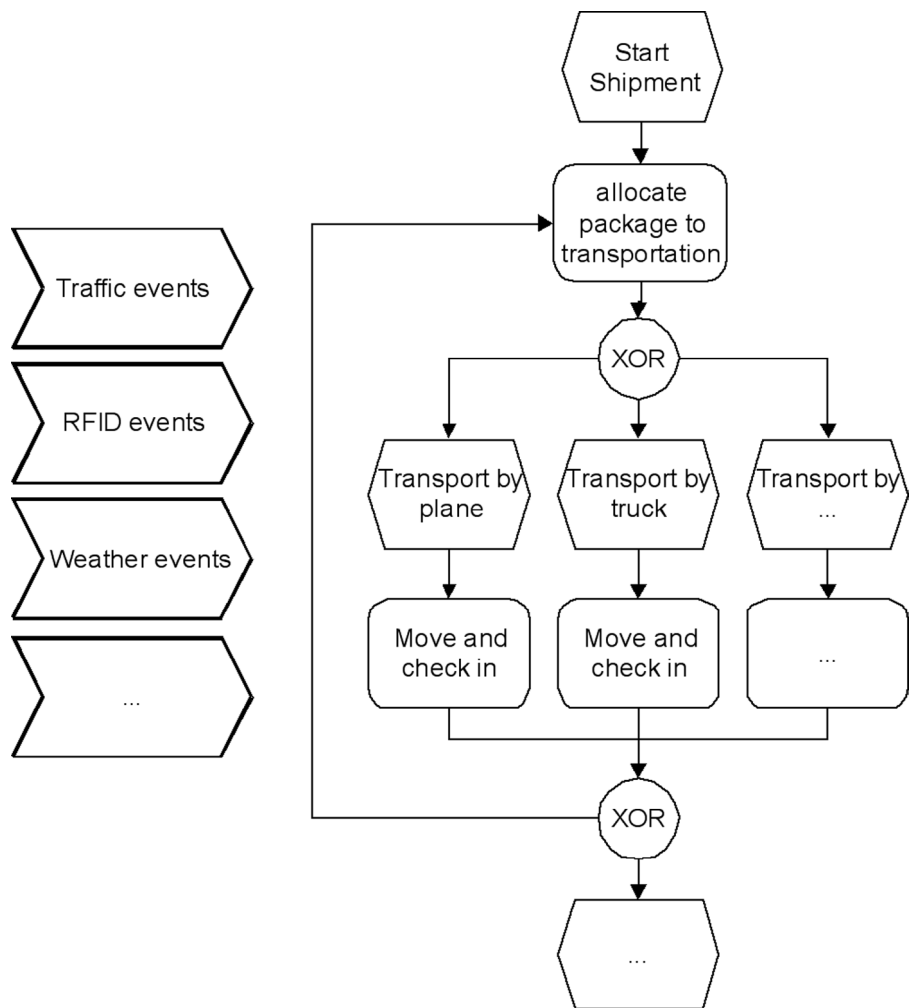


Abb. 2: Ausschnitt aus dem vereinfachten Paketzustellungsprozess.

Der Einfluss prozessexterner Ereignisse

Bei der Ausführung eines Prozesses treten Ereignisse auf. Bei EDBPM können – anders als bei BPMN – weitere Ereignisse außerhalb des modellierten Prozesses die Ausführung der Prozesse eines Unternehmens beeinflussen. Beim Prozess der Paketauslieferung könnte ein solches Ereignis ein Verkehrsstau auf einem vorher für die Tourenplanung festgelegten Streckenabschnitt sein. Der betroffene Abschnitt soll im nächsten Prozessschritt von einem Transportfahrzeug durchquert werden. Sobald das Fahrzeug auf der Strecke eintreffen würde, auf der die Verkehrsbehinderungen auftreten, erhöht sich die Wahrscheinlichkeit, dass die Paketzustellung nicht mehr in der vereinbarten Zeit möglich ist.

Zunächst muss geklärt werden, wie der Einfluss eines externen Prozessereignisses

gefunden werden kann. Dafür müssen zuerst sowohl die Zustandsinformationen des Prozesses als auch Informationen externer Ereignisse gesammelt und analysiert werden. Externe Prozesseinflüsse können während der Prozessausführung ununterbrochen in unregelmäßigen zeitlichen Abständen auftreten. Um eine Vorhersage treffen zu können, ob der Prozess bei seiner Ausführung erfolgreich ist oder ob ein Prozessabbruch auftreten wird, werden externe Ereignisse mit Prozessereignissen korreliert. Das Ergebnis der Korrelation ist eine Vorhersage, die – basierend auf zuvor definierten Kennzahlen, so genannten *Key Performance Indicators (KPI)* – getroffen wird. Durch die Mustererkennung von externen Prozesseinflüssen kann mittels CEP sowohl der Erfolg der Prozessausführung vorhergesehen werden, als auch in Echtzeit auf das sich anbahnende oder bereits erfolgte Versagen des Prozesses durch automatisierte



Vom Prozess erzeugtes Ereignis	Prozessexternes Ereignis
<ul style="list-style-type: none"> timeStamp: Identifizierer zur Angabe einer Reihenfolge der Ereignisse 	<ul style="list-style-type: none"> timeStamp: Identifizierer zur Angabe einer Reihenfolge der Ereignisse
<ul style="list-style-type: none"> itemId: Identifizierer des betroffenen Pakets 	<ul style="list-style-type: none"> itemId: Tatsächliches Gebiet des Ereignisses
<ul style="list-style-type: none"> processLoc: Identifizierer des Prozessschritts 	<ul style="list-style-type: none"> probability: Grad der Fehlerwahrscheinlichkeit
<ul style="list-style-type: none"> areaCode: Tatsächliches Gebiet des Ereignisses 	

Table 1: Grundsätzliche Attribute für die Ereignisverarbeitung des Paketzustellungsprozesses.

Aktionen und/oder Benachrichtigungen an bestimmte Rollen reagiert werden. In diesem *Enterprise Cockpit* können mehrere oder alle Prozessinstanzen unterschiedlicher Prozessstypen aggregiert und als Gesamtzustand eines Unternehmens angezeigt werden. Daneben besteht die Möglichkeit, Detailinformationen der Störungsursache anzuzeigen, z. B. kann im Prozess Paketauslieferung eine bestimmte Prozessinstanz ausgewählt und detailliert analysiert werden, in welcher Region Probleme auftreten und welche anderen Prozesse davon betroffen sind.

Ein logistischer Zustellungsprozess ist ein Prozess von nicht trivialer Komplexität, dessen zahlreichen Schritte bis zu seiner erfolgreichen Ausführung bereits dargestellt wurden. Viele Einflussfaktoren, die aber einen Einfluss auf den Erfolg der Zustellung haben, können beim Pro-

zessdesign nicht bedacht werden (z. B. Verkehrsstaus, Änderungen von Flugplänen, Unfälle und schlechte Wetterbedingungen). Im Folgenden wird konkret gezeigt, wie der Prozess bei seiner Ausführung Ereignisse an bestimmte Endpunkte versenden muss, damit dieser Zustellungsprozess überwacht werden kann. Diese wurden bei der Modellierung festgelegt. Zusätzlich werden prozessexternen Ereignisse an die Endpunkte versendet.

Für die Aggregation und Korrelation der Ereignisse werden mindestens die Informationen aus **Table 1** benötigt. Man bemüht sich seit einigen Jahren auch um eine Standardisierung der Informationsstruktur eines Ereignisformats für alle Ereignisquellen, z. B. das *Common Base Event (CBE)* als ursprünglicher Vorschlag von IBM und Cisco oder das *Webservice*

Event Format (WEF), standardisiert von OASIS 2005 (vgl. [IBM06]).

Die Vorhersage des Prozessverhaltens wird errechnet aus dem Zusammentreffen der Daten, d. h. einem vom Prozess erzeugten Ereignis sowie einem externen Ereignis und der dort angegebenen Fehlerwahrscheinlichkeit. Hierbei erfolgt ein Abgleich der Streckengebiete eines externen Ereignisses mit einem Prozessereignis. Dabei wird zuerst überprüft, ob es sich um das nachfolgende Streckengebiet handelt, das vom Fahrzeug im nächsten Schritt durchquert wird. Solche mögliche Ereigniskorrelationen zu komplexen Ereignissen können in einer Grammatik als EBNF-Notation wie folgt ausgedrückt werden:

```

Komplexes_Prozess_Problem = Prozessschrittereignis |
{[Prozessschrittereignis | Externes_Ereignis]}
Prozessschrittereignis = "Prozessschritt gestartet" | "Prozessschritt beendet" |
"Service gestartet" ...
Externes_Ereignis = Verkehrsstau | Wetter | ...
Verkehrsstau = "Highway No.1" | "Route 66" ...
    
```

Die Grammatik besagt, dass sich ein komplexes Prozessproblem aus einem oder mehreren Ereignissen eines Prozesses sowie – nicht notwendigerweise – einem externen Ereignis zusammensetzt. Ein Prozessschrittereignis beinhaltet z. B. Informationen darüber, ob ein Prozessschritt oder ein zugehöriger Service gestartet oder beendet wurde. Externe Ereignisse liefern z. B. Informationen über Verkehr und Wetterlage. Betrachtet man das Verkehrsereignis genauer, so beinhaltet es Informationen über Strecken, auf denen der Stau auftritt.

Abbildung 3 visualisiert diese Hierarchie. Dabei liegt das Stauereignis auf der nie-

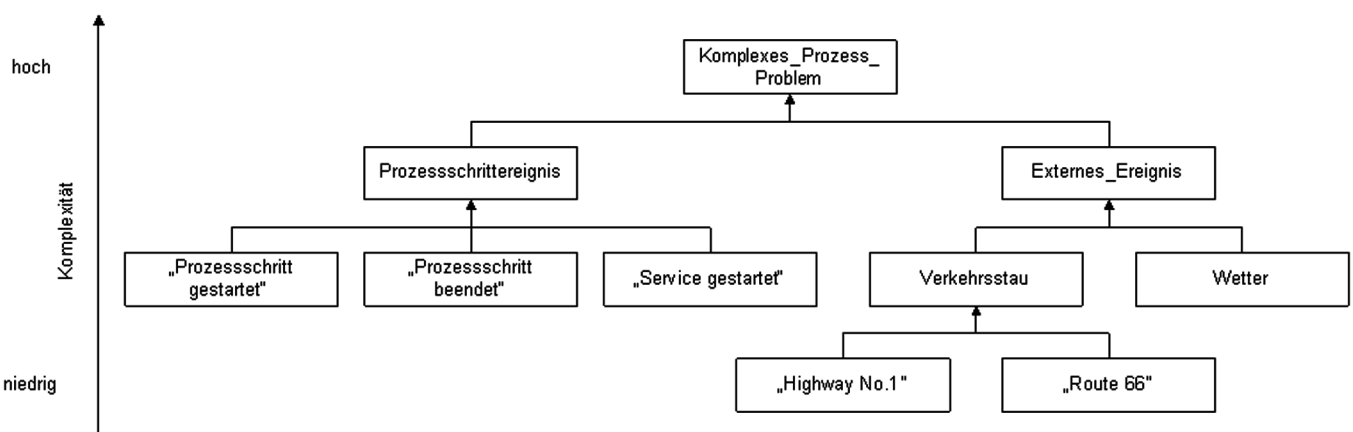


Abb. 3: Hierarchische Darstellung der EBNF-Ereignis-Notation.

drigsten und das komplexe Prozessproblem auf der höchsten Komplexitätsebene.

Die Logik der Ereignisverarbeitung wird im Detail derzeit in einer SQL-ähnlichen oder einer völlig proprietären EPL ausgedrückt. Solche Programme können sehr umfangreich und kompliziert werden. Sie erfordern bei Prozessen, wie im Beispiel angedeutet, einen Spezialisten – den erwähnten Ereignis-Modellierer.

Weitere aktuelle Projekte und Studien

Eine andere Studie und ein PoC entstanden in Zusammenarbeit mit der früheren norisbank und der heutigen TeamBank (vgl. [Bra07]). Auf Basis eines Online-Kredit-systems werden täglich tausende von Kreditanträgen gestellt. Viele der Anträge werden bei dem Online-Prozess abgebrochen. Daraus ergibt sich ein erheblicher Verlust, den man minimieren möchte. Eine entsprechende Überwachung der Antragsprozesse kann wegen der großen Anzahl nur automatisiert erfolgen. Dazu müssen typische Ereignismuster entworfen werden, die auf einen drohenden Abbruch des

Antragsprozesses hindeuten. Jedem Ereignismuster ist dann ein Prozess oder eine Aktion zuzuordnen, die möglichst in Echtzeit ausgeführt werden und den Kunden halten soll. Zu diesem Zweck muss der Ereignis-Modellierer die entsprechenden Ereignistypen identifizieren. Da sich auch bei diesem PoC zeigte, dass entsprechende Ereignisse nicht schon im implementierten Prozess verfügbar waren, mussten auch bei diesem Anwendungsfall zusätzliche Ereignisse in dem Kreditantragsprozess generiert werden (vgl. ausführlich [Bra07]).

Bei einem anderen Use-Case sollte eine Warnung an einen Angestellten der Bank vom EDBPM-System geschickt werden, falls der aggregierte Verlust wegen abgebrochener Kreditanträge einen bestimmten Schwellenwert erreicht. Es wäre nicht sinnvoll, bei jedem Antragsabbruch eine Warnung zu senden, da zu viele Warnungen zum so genannten *V.I.R.T.*-Problem (*Valuable Information at the Right Time*) führen würden (vgl. [Hay07]). Auch wie man damit im Rahmen von CEP-Anwendungen geeignet umgeht, ist noch zu klären. ■

Literatur & Links

[Amm07] R. v. Ammon, C. Silberbauer, C. Wolff, Domain Specific Reference Models for Event Patterns – for Faster Developing of Business Activity Monitoring Applications, in Proc. of VIPSI 2007

[Bra07] H.-M. Brandl, D. Guschakowski, Complex Event Processing in the context of Business Activity Monitoring, Diplomarbeit, 2007, siehe: www.citt-online.de/downloads/Diplomarbeit_BaGu_Final.pdf

[Bru03] Bruce Silver Associates, FileNet P8 – Event-Driven Business Process Management, in: Industry Trend Reports June 2003, siehe: www.ecm-unverzichtbar.de/Repository/48/files/69_EvBPM.pdf

[CEP06] 1st CEP-Symposium, Hawthorn/NY, March 2006, siehe: complexevents.com/?page_id=87

[CEP08] Use cases of CEP applications, siehe: complexevents.com/?cat=16

[Dan07] K. Danielsson, G. Trotta, Key Requirements for Event-Driven BPM and SOA, siehe: www.ebizq.net/topics/bpm/features/8700.html

[Emm08] C. Emmersberger, F. Springer, Event Driven Business Process Management. An evaluation of the approach of Oracle and the Sopera Open Source Framework, Diplomarbeit 2008, siehe: www.citt-online.de/downloads/EmmSpr_Diplomarbeit_Final.pdf

[EPTS07] Event Processing Technical Society, siehe: www.ep-ts.com/component/option,com_frontpage/Itemid,1/

[Hay07] F. Hayes-Roth, Value Information at the Right Time (VIRT): Why less volume is more value in hastily formed networks, 2007, siehe: www.nps.edu/cebrowski/Docs/VIRTforHFNs.pdf

[IBM06] IBM, Best Practices for the Common Base Event and Common Event Infrastructure 2006, siehe: download.boulder.ibm.com/ibmdl/pub/software/dw/autonomic/books/cbeppractice/index.htm

[Luc02] D. Luckham, The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems, Addison-Wesley 2002

[OMG06] OMG/BPMI, Business Process Modeling Notation (BPMN) Information, siehe: www.bpmn.org/

[Ran06] V. Ranadive, The Power to Predict: How Real Time Businesses Anticipate Customer Needs, Create Opportunities and Beat the Competition, McGraw-Hill Professional, 2006

[Sch06] R. Schulte, zitiert in Computerwoche v. 18.7.2006, Artikel „CEP verheißt den Blick in die Zukunft“, siehe: www.computerwoche.de/heftarchiv/2006/29/1215287/