

SOA MEETS EPK – SERVICE-ORIENTIERTE GESCHÄFTSPROZESSE DURCHGÄNGIG MODELLIEREN

Die verschiedenen Ansätze für Geschäftsprozessmodellierung unterscheiden sich zum Teil erheblich in Zielsetzung und Detaillierungsgrad, da sie aus unterschiedlichen Kontexten entstanden sind. Gerade bei service-orientierten Architekturen fällt bei den relevanten Modellierungstechniken eine systematische Lücke zwischen technischer und semantischer Beschreibung auf. Um diese Lücke, die ein Hindernis für die erfolgreiche Modellierung service-intensiver Prozesse darstellt, zu schließen, schlagen wir eine Erweiterung des Formalismus der ereignisgesteuerten Prozessketten vor. Damit lässt sich eine Brücke zwischen den EPK und dem technischen Standard BPMN schlagen.

In den letzten zehn Jahren wurden eine ganze Reihe von Ansätzen zur grafischen Modellierung von Geschäftsprozessen vorgestellt. Einige davon konnten sich inzwischen zu Standards bzw. Quasi-Standards entwickeln. Eine solche Modellierung ist zwar nicht für alle Abläufe innerhalb eines Unternehmens oder einer anderen Organisation erforderlich. Zur Anwendung von formalen Analysemethoden mit dem Ziel einer Verifikation oder Optimierung der Prozesse sind solche formalen Modelle jedoch unverzichtbar.

Angesichts heutiger service-orientierter Architekturen (SOA) mit loser Kopplung, aber starker Prozessfokussierung, ist es nicht mehr ausreichend, die Modellierung der Geschäftsprozesse auf ein Unternehmen oder gar nur eine Organisationseinheit zu beschränken. Die tatsächlichen Prozesse reichen längst über Unternehmensgrenzen hinweg. Daher müssen auch deren Abbildungen in der IT in der Lage sein, sich übergreifend und in Form von loser Kopplung mit anderen Systemen zu integrieren.

Aus unserer Sicht ist es dafür auch notwendig, die Formalismen der Geschäftsprozessmodellierung zu erweitern, und zwar um den Begriff „Service“ (oder „Dienst“). Wir schlagen daher vor, eine weit verbreitete Notation zur Geschäftsprozessmodellierung, die ereignisgesteuerten Prozessketten (EPK), um Elemente für Services zu erweitern. Im Folgenden wollen wir diese Elemente näher erläutern, Beispiele für deren Anwendung in Szenarios mit Service-Kompositionen geben und diesen Ansatz mit anderen formalen Dienstdarstellungen wie BPMN vergleichen.

Bei der Geschäftsprozessmodellierung ist zwischen textuellen Notationen wie BPEL, ebXML oder XPD L einerseits und grafischen Notation wie UML, BPMN oder EPK andererseits zu unterscheiden. Neben dem „Universalwerkzeug“ UML sind es dabei vor allem BPEL, BPMN und EPK, die inzwischen die größte Bedeutung erlangt haben. BPMN, die *Business Process Modelling Notation (BPMN)* [Whit2004], wurde 2004 bei IBM entworfen und ist inzwischen zu einem OMG-Standard geworden (www.omg.org). Komplementär dazu ist BPEL, die *Business Process Execution Language (BPEL)* [AAAs2007], die bei der Organization for the Advancement of Structured Information Standards (OASIS, www.oasis-open.org) entwickelt wurde und auch von dieser koordiniert wird. BPEL ist das heute am meisten genutzte Format für Geschäftsprozessmodellierung in XML. Die OMG und OASIS sind beide herstellernerneutrale Vereinigungen. Sie verstehen sich nicht als Konkurrenten, sondern arbeiten eng miteinander zusammen und ergänzen sich gegenseitig. Dementsprechend sind auch BPMN und BPEL nicht als Alternativen mit gleicher Zielsetzung, sondern als komplementäre Konzepte zu verstehen.

Ein weiterer bedeutender Vertreter einer rein grafischen Notation sind die <I>ereignisgesteuerten Prozessketten (EPK)</I>. Dieses Konzept wurde gemeinsam von SAP und der Universität des Saarlands seit 1992 als halbformale Beschreibung von Geschäftsprozessen und Arbeitsabläufen entwickelt (vgl. [Sche2006]). Diese werden vom

▶ die autoren



Stefan Huth

(E-Mail: stefan.huth@prodyna.de) ist IT-Consultant bei der PRODYNA GmbH. Bereits während seines Studiums der Informatik beschäftigte er sich mit der Thematik der Softwareentwicklung unter Java. Zuletzt arbeitete er als technischer Angestellter an der Hochschule Coburg an einem wissenschaftlichen Projekt in den Bereichen MDA und SOA.



Dr. Thomas Wieland

(E-Mail: wieland@c-fis.de) ist Professor für Informatik an der Hochschule Coburg und Geschäftsführer des c-fis – Centrum für innovative Softwaresysteme GmbH. Das c-fis unterstützt Unternehmen bei der Planung und Realisierung von Softwareprojekten und berät in Fragen aktueller Technologien wie SOA, Open Source oder Pervasive Computing.

größeren zum feineren Niveau modelliert. Dabei kann eine Ebene in weitere Einzelprozesse aufgeteilt werden, die dann separat in größerer Detailtiefe dargestellt werden. Die Elemente des EPK-Formalismus sind Funktionen, Ereignisse und der Kontrollfluss. In der eEPK, der erweiterten EPK, finden sich zudem Organisationseinheiten, Dokumente und Informationsobjekte.

Die ereignisgesteuerten Prozessketten werden verwendet, um einen Prozess aus der Geschäftsperspektive zu beschreiben. Sie können so ein Mittel zur individuellen Anpassung von prozessorientierten Informationen sein. Das EPK-Konzept bildet auch die Basis der ARIS-Plattform, wie sie von der IDS Scheer AG vertrieben wird.

Geschäftliche Relevanz von Services

Für die reine Modellierung eines geschäftlichen Ablaufs ohne die Perspektive auf ein unterstützendes IT-System ist der klassische

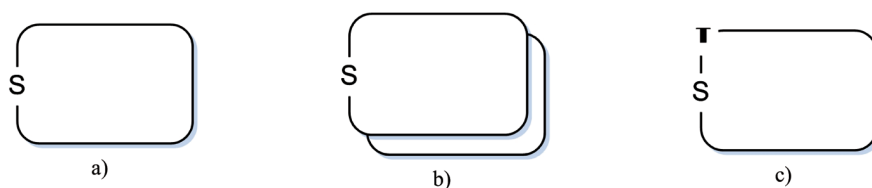


Abbildung 1: Die neuen Elemente in der sEPK-Notation: a) Synchroner Service, b) asynchroner Service, c) Service-Template

Ansatz immer noch sinnvoll und nützlich. Doch diese Situation ist immer seltener anzutreffen und zwar aus gutem Grund: Um Geschäftsprozesse heute kosteneffizient und flexibel etablieren zu können, ist deren Unterstützung durch perfekt zugeschnittene IT-Systeme zwingend erforderlich. Immer wenn Fachexperten und IT-Experten gemeinsam daran arbeiten, ein Modell zu entwickeln, zu verfeinern und zu optimieren, stehen die IT-Fachleute vor der Aufgabe, die Konsequenzen in Ihre Überlegungen einzubeziehen, die sich aus einer service-orientierten Realisierung ergeben. In diesem Sinne hebt sich die „SOA-Geschäftsprozessmodellierung“ deutlich von der klassischen Vorgehensweise ab.

In einem EPK-Modell könnte nun jede Funktion durch einen Service realisiert werden. Ein solcher Service ist aber nicht nur ein beliebiges Artefakt der Programmierung. Bereits die Entscheidung, welche Art von Service genutzt werden soll, ist genauso technischer wie strategisch-ökonomischer Natur. Denn neben rein internen Services kann es auch solche geben, die auf externe Informationen oder Dienstleistungen zugreifen und daher auch extern aufgerufen werden müssen. Beispiele sind Lieferinformationen von Zulieferern oder Überprüfung von Kreditkartenangaben. Wenn man jedoch externe Dienste einbindet, erzeugt man damit Abhängigkeiten und erhöht die Komplexität des Managements der IT-Dienstleistungen und der Softwarekomponenten. Auf der anderen Seite kann es auch eigene hochentwickelte Dienste geben, die so reif und vielseitig sind, dass sie selbst anderen angeboten werden können – ggf. sogar gegen Gebühr. Das ergibt dann weitere technische und ökonomische Implikationen.

Daraus wird offensichtlich, dass selbst in einem Prozessmodell mit rein geschäftlichem Fokus diese Aspekte nicht außer Acht gelassen werden dürfen, sondern sich auf die eine oder andere Weise ausdrücken lassen sollten. Dafür ist eine Erweiterung der Notationsmöglichkeiten erforderlich.

sEPK – service-orientierte ereignisgesteuerte Prozessketten

Die von uns vorgeschlagene Erweiterung, die wir sEPK nennen wollen für „service-orientierte ereignisgesteuerte Prozessketten“, fügt neue Elemente für eine service-orientierte Geschäftsprozessmodellierung hinzu. Um das Lesen und unmittelbare Verstehen solcher erweiterten Prozessmodelle zu erleichtern, gibt es verschiedene Symbole für Services, deren Eigenschaften und Implikationen. Die Symbole der sEPK können wie die anderen Elemente der eEPK verwendet werden. Sie bieten lediglich zusätzliche Möglichkeiten für die Modellierung von Prozessen mit Services.

Eine Intension hinter der Schaffung der neuen Symbole war, dass der Modellierer damit sofort die Struktur des Geschäftsprozessmodells erkennen kann, ohne dass er dazu in den Kommentaren nachschlagen muss. Mit einem Blick lassen sich so Prozessereignisse, die von Funktionen herühren, von synchronen oder asynchronen Dienstresultaten unterscheiden.

Das erste Symbol ist eine Abwandlung des Funktionssymbols der EPK-Notation. Es stellt eine Funktion eines synchron aufgerufenen Service dar (**Abbildung 1a**). Man setzt es wie eine normale Funktion in einem Geschäftsprozess innerhalb einer Prozesskette ein. Ein synchroner Service wird üblicherweise als entfernter Prozeduraufruf realisiert. Um mit dem nächsten Ereignis oder Service fortzufahren, muss die aufgerufene Funktion geendet haben.

Abbildung 1b zeigt das Symbol für einen asynchronen Service. Genau wie der synchrone Service kann der asynchrone anstelle einer normalen Funktion verwendet werden. Der einzige Unterschied ist hierbei, dass der asynchrone Service nicht geendet haben muss, um zum nächsten Schritt weiterzugehen. Der Service kann daher aufgerufen und anschließend „vergessen“ werden. Diese Eigenschaft sollte jedoch vom Modellierer des Prozesses berücksichtigt werden. Denn die Art des Aufrufs des

Services hinter einer EPK-Funktion hat eine direkte Auswirkung auf das Modell. Bei einem optimierten Modell sollte der asynchrone Charakter einer Funktion berücksichtigt und aktiv genutzt werden, zum Beispiel durch Ausführung einer anderen Funktion parallel dazu in der Zwischenzeit, sodass der gesamte Workflow nicht durch das Warten auf die Beendigung des Service blockiert wird.

Das SOA-Konzept verspricht, dass dementsprechend gebaute Softwaresysteme nicht mehr monolithisch und fest „verschraubt“ sind, sondern dynamisch und agil. So ist es beispielsweise möglich, dass ein Service von einem IT-System aufgerufen wird, der zum Zeitpunkt des Entwurfs dieses Systems noch völlig unbekannt war, der sich höchstens an eine vordefinierte Schnittstelle halten muss. Zur Laufzeit setzt sich das System mit einem Verzeichnisdienst in Verbindung, fragt nach einem Service, der eine bestimmte Aufgabe erfüllt und dafür eine bestimmte Schnittstelle unterstützt, und erhält die technischen Daten für einen oder mehrere vormals unbekannte Services. Solche Verzeichnisdienste können beispielsweise von branchenspezifischen Marktplätzen für Zulieferer angeboten werden. Aber auch ein Enterprise Service Bus als Vermittlung in einer unternehmensweiten service-orientierten Architektur kann als Verzeichnisdienst auftreten.

Für die Geschäftsprozessmodellierung kann es somit zu der Situation kommen, dass Funktionen in einem Modell eingefügt werden, die nicht nur völlig extern sind, also außerhalb der Kontrolle der jeweiligen Organisation oder Organisationseinheit, sondern die auch noch zum Zeitpunkt der Erstellung des Modells unbekannt sind. Um dies auszudrücken, enthält die sEPK eine der wichtigsten Neuerungen: das Service-Template-Symbol (**Abbildung 1c**). Es wird nur über einen Namen spezifiziert. Das Template kann sowohl für einen synchronen als auch für einen asynchronen Service stehen. Die physikalische Instanz dieses Services wird dann erst zur Laufzeit bestimmt, z. B. durch Nachschlagen bei einem entsprechenden Verzeichnisdienst.

Neben den Funktionen benötigt die service-orientierte Geschäftsprozessmodellierung noch einige besondere Ereignisse. Das erste neue Element ist das Ausnahme-Ereignis („exception“, **Abbildung 2a**). Es weist darauf hin, dass eine Ausnahme von einem Service geworfen werden könnte, die zu behandeln ist.



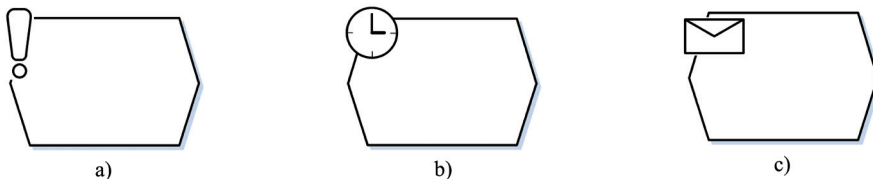


Abbildung 2: Ereignisse in der sEPK-Notation: a) Ausnahme-Ereignis, b) Timeout-Ereignis, c) Nachricht-erhalten-Ereignis

Daneben wird noch das Timeout-Ereignis als neues Element eingeführt (Abbildung 2b). Es wird verwendet für Timeouts, die im Zusammenhang mit Services auftreten könnten. Das kann beispielsweise der Fall sein, wenn ein Service aufgerufen wird und nicht innerhalb einer vorgegebenen Zeit antwortet. Es kann aber auch für Fälle verwendet werden, in denen Funktionen von Mitarbeitern ausgeführt werden müssen und die Situation eines versäumten Termins explizit berücksichtigt werden soll. Dies kann etwa für automatisierte Workflow-Systeme von Bedeutung sein. Bei Services treten Timeouts zwar häufiger bei synchronen Diensten auf. Sie sind jedoch auch bei asynchronen Services möglich.

Als weiteres neues Element wollen wir das Nachricht-erhalten-Ereignis (Abbildung 2c) vorschlagen. Es stellt ein Ereignis dar, das verwendet wird um auszudrücken, dass eine Nachricht von einem anderen System erwartet wird, die behandelt werden sollte. Ein typischer Fall ist die Rückgabennachricht eines asynchronen Dienstes. Diese Art von Ereignis kann beispielsweise auch bei der Realisierung mittels einer nachrichtenorientierten Middleware einge-

setzt werden. Wie andere Ereignisse auch kann es sowohl eine Prozesskarte starten als auch inmitten eines Workflow stehen. Die Rückantwort eines asynchronen Dienstes sollte explizit modelliert werden, da es sich dabei ja nicht um ein Ereignis handelt, das von einer Person oder einer Organisationseinheit ausgelöst wird, sondern durch ein automatisiertes technisches System. Daher erscheint es sinnvoll und ratsam, diese besondere Eigenschaft eines derartigen Ereignisses im Modell widerzuspiegeln. Letztlich könnten ganze Prozessketten durch ein solches Ereignis gestartet werden.

Attribute für Symbole

Einige Eigenschaften der Dienste lassen sich indessen nicht so einfach durch grafische Symbole ausdrücken oder würden die Grafik zu unübersichtlich machen. So ist beispielsweise die Organisationseinheit, die für den Service verantwortlich zeichnet, für die Prozessmodellierung nicht besonders relevant. Aber es handelt sich auch nicht um eine Information mit rein technischem Bezug – derartige Dinge wären sicherlich außerhalb des Rahmens der sEPK. Die

Zuordnung von Verantwortlichkeiten ist zweifelsohne eine geschäftliche Entscheidung. Sie ist jedoch für die Umsetzung des Prozesses wichtiger als für die grundlegende Definition des Workflow.

Wir schlagen daher vor, jede Funktion (normalerweise nur die Service-Funktionen) in einer sEPK mit einigen optionalen Attributen zu versehen, die zur Angabe von mehr Details genutzt werden können, als sich mit der grafischen Notation ausdrücken lassen. Diese sind in Tabelle 1 aufgelistet.

Modellierung mit asynchronen Diensten

Beim Modellieren eines Workflows können synchrone und asynchrone Services nacheinander oder parallel ausgeführt werden. Abbildung 3a zeigt, wie die beiden Arten von Services parallel genutzt werden können. Wird der synchrone Service verwendet, so wird die Kette normal nach der Beendigung der Funktion fortgesetzt. Falls der asynchrone Service aufgerufen wird, setzt sich der Prozess fort ohne zu wissen, ob und wann die Funktion endet. Um sicherzustellen, dass die Funktion erfolgreich beendet werden konnte, sollte daher bei der Modellierung ein entsprechendes Ereignis eingefügt werden.

Es ist auch möglich so lange abzuwarten, bis der asynchrone Dienst antwortet. Doch der übliche Anwendungsfall einer asynchronen Kommunikation beinhaltet, dass die Funktion aufgerufen wird und die Antwort zu einem späteren Zeitpunkt eintrifft. Daher ist dies ein entscheidender Aspekt bei der Modellierung mit Services, besonders dann, wenn es parallele Wege in der Prozesskette gibt. Betrachten wir daher noch das Zusammengehen der beiden Äste genauer. Wenn zwei oder mehr Zweige miteinander verbunden sind, bedeutet das, dass der gesamte Prozess so lange warten muss, bis alle beendet sind, d. h. bis eine Antwort eintrifft oder ein Timeout-Ereignis eintritt. Bei synchronen Diensten, die zumeist entfernte Prozeduraufrufe verwenden, wird die Kette fortgesetzt, wenn die Funktion endet. Bei asynchronen Diensten – und ebenfalls bei einer typischen Kommunikation mittels nachrichten-orientierter Middleware – ist der Funktionsaufruf mit dem Absenden der Nachricht beendet. Die Antwort hängt dann vom aufgerufenen Service ab. Im schlimmsten Fall, falls die Antwort ganz ausbleibt, kommt so der gesamte Workflow durch das Warten auf eine Nachricht zum Erliegen. In dieser Situation sollte ein Nachricht-erhalten-

Attribut	Beschreibung
Responsibility	Dieses Attribut beschreibt, welche Organisationseinheit (Abteilung, externer Partner, externe Dienstleister) verantwortlich für die Bereitstellung und Wartung des Dienstes ist.
Security	Dieses Attribut führt die Sicherheitsanforderungen (Autorisierung, Authentifizierung, Verschlüsselung) auf, die für den Aufruf dieses Dienstes gelten.
Input	Ein oder mehrere Eingaben können für einen Dienst definiert werden. Diese lassen sich explizit durch Daten/Material-Objekte zeichnen oder in textueller Form in diesem Attribut vermerken, wobei hier auch weitere Details über die Vorbedingungen der Eingabeobjekte hinterlegt werden können. Es sollte aber nur eine der beiden Möglichkeiten genutzt werden, um Mehrdeutigkeiten auszuschließen.
Output	Jede Funktion kann eine oder mehrere Ausgaben bereitstellen. In diesem Attribut lässt sich beschreiben, welche Art von Objekt jedes Ausgabeelement darstellt. Zudem können Nachbedingungen der Ausgabeobjekte angeführt werden.

Tabelle 1: Optionale Attribute für sEPK-Funktionen

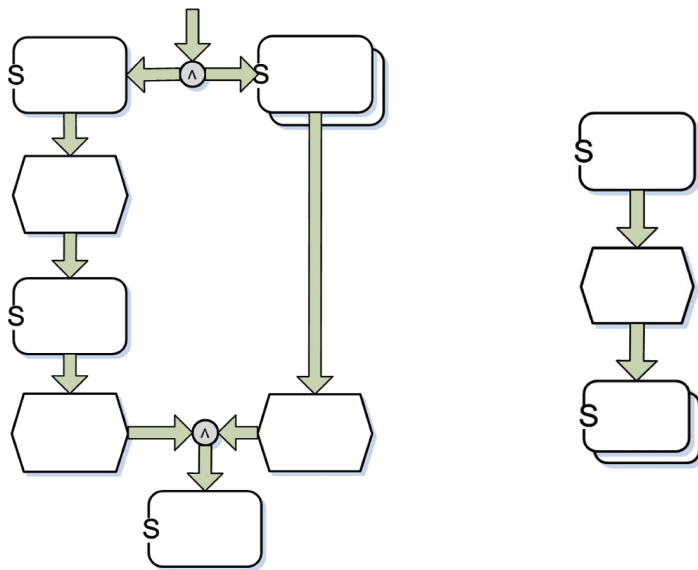


Abbildung 3: Workflow mit synchronen und asynchronen Services: a) Parallele Abarbeitung, b) Ende des Workflow mit einem asynchronen Service

Ereignis eingesetzt werden, das der Service auslöst, um über seinen erfolgreichen Abschluss zu informieren.

Bei asynchronen Diensten kommt gelegentlich auch das „fire and forget“-Muster vor, bei dem ein Service aufgerufen wird und sich die aufrufende Kette nicht für das Resultat interessiert. Die Konsequenz davon ist letztlich, dass also ein Workflow auch mit dem Aufruf eines Service enden kann – ohne zu wissen, ob alle Aufrufe aus der Kette auch erfolgreich abgearbeitet werden konnten (**Abbildung 3b**). Ein Beispiel dafür ist das Verschicken einer E-Mail an einen Kunden, wenn ein Bestellvorgang abgeschlossen ist. Diese Modellierungsvariante steht im Widerspruch zu den Bildungsregeln von klassischen EPKs. Daher ist sie nur in seltenen Ausnahmefällen zu verwenden. Sie zeigt indessen, dass das SOA-Paradigma über das Potenzial verfügt, die Praktiken der Geschäftsprozessmodellierung auch in unkonventionelle Richtungen zu erweitern.

Für weitere, umfangreichere Beispiele für sEPK-Modelle sei auf [HuWi2007] verwiesen.

sEPK im Vergleich zu BPMN

Man könnte die sEPK-Notation als eine Abbildung der BPMN-Elemente in den EPK-Formalismus betrachten. Wie oben erläutert erlangen diese Erweiterungen Bedeutung durch den Einfluss der Serviceorientierung auf Geschäftsprozesse und deren Modellierung. Die Ähnlichkeit zwischen sEPK und BPMN ist daher nicht zufällig, sondern durchaus beabsichtigt, da es sich bei BPMN um einen weit verbreite-

ten Standard für Geschäftsprozessmanagement handelt.

BPMN ist verhältnismäßig stark technikorientiert und verfügt nicht über Elemente wie Organisationseinheit oder Informationsobjekt, die wesentliche Bestandteile der eEPK sind. Auf der anderen Seite verfügt aber BPMN über einige Merkmale, die in eEPK und sogar noch in sEPK fehlen, zum Beispiel bei der Modellierung von eventbasierten Gateways in einem Workflow. Der BPMN-Standard beinhaltet dafür ein eigenes Darstellungsformat. Selbst mit sEPK muss zur Modellierung einer solchen Verzweigung eine umfangreichere Ereignisverknüpfung angelegt werden. Dies ist zwar eine Einschränkung im Vergleich zu BPMN. Eine Erweiterung des EPK-Formalismus um derartige Möglichkeiten hätte jedoch eine zu große, umwälzende Änderung und eine Abkehr von den bekannten Bildungsregeln für EPK bedeutet. Damit wären die Abwärtskompatibilität und letztlich wohl auch die Akzeptanz gefährdet gewesen. Daher verzichtet der sEPK-Vorschlag auf solche Formate.

Eine weitere Abweichung zwischen den beiden Notationen liegt in der Unterteilung der sonstigen Ereignisse bei BPMN. Dort wird zwischen Start-, Zwischen- und Endereignissen unterschieden. Jeder der drei Typen wird durch ein eigenes Symbol gekennzeichnet. In der sEPK-Notation gibt es nur eine Art von Ereignis und keine Unterscheidung nach dessen Position im Workflow. Auch die Zahl und Vielfalt der Attribute ist in BPMN wesentlich höher.

Eine Gegenüberstellung der sEPK- und BPMN-Modelle eines umfangreicheren Prozessbeispiels ist in [HuWi2007] zu finden.

Schrittweise Verfeinerung in den Prozessmodellierungen

Obwohl die Ergänzungen des vorgeschlagenen sEPK-Formalismus gegenüber den klassischen eEPK nur eine Erweiterung um einige grafische Bausteine und Notationsregeln bestehen, sind es doch gerade diese Elemente, die für einen serviceorientierten Workflow entscheidend sind. Zudem hat diese Erweiterung den Vorteil, dass sie sich harmonisch mit den üblichen Elementen integriert. Die sEPK als Erweiterung der eEPK verfügt insgesamt – verglichen mit anderen grafischen Notationen wie BPMN – über mehr Funktionselemente, wie Organisationseinheiten oder Information/Material- und Ressourcen-Objekten. Somit lassen sich auf diese Weise mehr Facetten des Workflows abbilden.

Da die neuen Elemente optional sind, entsteht durch deren Vermeidung aus der sEPK wieder die eEPK. Falls es dagegen notwendig und sinnvoll ist, Services als Funktionen anzubieten oder Funktionen als Services zu kennzeichnen, können die neuen Symbole verwendet werden.

Mit Hilfe der verschiedenen BPM-Notationen lässt sich zusammen genommen eine Werkzeugkette aufbauen, die von der reinen Geschäftsprozessanalyse bis zum Laufzeitsystem der entsprechenden IT-Komponenten reicht. Zunächst analysieren Fachexperten die realen Prozesse und erzeugen dazu formale Beschreibungen, z.B. unter Verwendung der EPK-Notation. Oft ist dieser Schritt auch mit einer Optimierung dieser Prozesse verbunden. Diese Informationen können dann verwendet werden, um die rechnergestützten Abbildungen direkt zu modellieren, im Falle eines SOA-Ansatzes mittels BPMN.

Doch diese beiden Formalismen gehen von sehr unterschiedlichen Sichtweisen auf die Geschäftsprozesse und die Informationssysteme aus. Während die EPKs sich auf die Geschäftslogik und die organisatorischen Randbedingungen konzentrieren, sind BPMN-Modelle stärker technologiezentriert und arbeiten daher auf einem anderen Abstraktionsniveau. Es fehlt also eine Verbindung zwischen diesen beiden Modellen. Diese Verbindung stellt der oben vorgestellte Formalismus der serviceorientierten EPKs dar. Fügt man diese Modellierungsmöglichkeit in die Werkzeug-



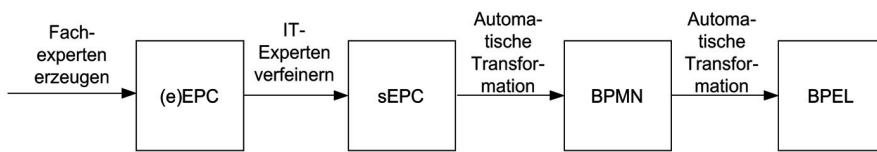


Abbildung 4: Werkzeugkette von der Prozessanalyse bis zur Prozessablaufumgebung

kette ein, so erhält man eine relativ bruchfreie schrittweise Verfeinerung von der Geschäftsprozessanalyse bis zur Ablaufumgebung (Abbildung 4).

Die anfängliche formale Prozessbeschreibung mittels EPKs (oder eEPKs) wird nach wie vor von Fachexperten erledigt. Mit dem Ziel einer service-orientierten Architektur übernehmen dann IT-Experten dieses Modell und verfeinern bzw. konkretisieren es. Dabei werden Funktionen daraufhin überprüft, ob sie als Services realisiert werden können oder müssen, es werden synchrone und asynchrone Kommunikationspfade hinzugefügt und mögliche Ausnahme sowie Nachricht-erhalten-Ereignisse integriert oder modifiziert. Das Ergebnis ist dann immer noch ein Geschäftsprozessmodell, das jedoch die nichtfunktionale Anforderung erfüllt, dem SOA-Paradigma zu genügen.

Der entscheidende Vorteil dieses Zwischenschritts ist, dass mit dieser Verfeinerung und Klarstellung das EPK-Modell auf ein technologie-orientiertes Niveau gebracht wird – und das ohne einen vollständigen Bruch der Methodik. Auf diese Weise kann eine Abbildung von der

geschäftlichen auf die technische Realisierung hergestellt werden. Auf der anderen Seite basieren die sEPK auf denselben Bausteinen wie BPMN, sodass eine automatische Transformation von sEPK nach BPMN möglich wird. Der nächste Schritt, die weitere automatische Transformation von BPMN in die Ausführungssprache BPEL ist bereits Forschungsziel verschiedener Arbeitsgruppen, bietet jedoch auch noch einige systematische Herausforderungen ([ODAH2006], [OADH2006], [ReMe2006]).

Fazit

Die service-orientierten ereignisgesteuerten Prozessketten verbinden die Vorteile der gebräuchlichen eEPK-Notation mit den Eigenschaften des technischen Formats BPMN. Dabei wird der eEPK-Formalismus um die Elemente synchroner und asynchroner Service, Template-Service, Ausnahme-Ereignis, Timeout-Ereignis sowie Nachricht-erhalten-Ereignis erweitert. Die Service-Elemente können dabei noch um Attribute angereichert werden.

Damit eröffnet sich die Möglichkeit, sowohl klassische Workflows als auch

Prozesse für service-orientierte Architekturen zu modellieren. Da sich die BPMN-Definition noch im Fluss befindet und es bereits jetzt immer noch einige wenige Elemente gibt, die kein Pendant in der sEPK haben, ist als weiterer Ausbau eine kontinuierliche Aktualisierung sowie eine noch stärkere Abdeckung der BPMN-Elemente im Rahmen einer noch etwas umfangreicheren sEPK geplant.

Der nächste Schritt ist dann, ein grafisches Werkzeug zu schaffen, mit dem sich sEPK-Modelle in BPMN und diese weiter in eine Ausführungssprache wie BPEL transformieren lassen, wie in Abb. 4 gezeigt. Mit einem solchen Werkzeug kann der Benutzer Prozesse auf der Basis bestehender oder geplanter Services modellieren und die erzeugten BPEL-Daten unmittelbar in eine Ablaufumgebung integrieren (wie auch in [Whit2005] vorgeschlagen).

Die wissenschaftliche interessanteste Idee an diesem Werkzeug ist die Transformation von Modellen zwischen verschiedenen Notationen während der Bearbeitung. So könnte ein Prozess von sEPK nach BPMN ohne Informationsverlust umgewandelt werden. Das ist möglich, da sich die sEPK-Elemente direkt auf die BPMN-Gegenstücke abbilden lassen. Somit sind alle Bestandteile für ein service-orientiertes Geschäftsprozessmanagement gegeben. Die nächste Herausforderung ist dann die Transformation der Modelle nach BPEL, die sicher nur teilweise automatisch erfolgen kann. ■

Referenzen

- [AAAs2007] Alves, A.; Arkin, A.; Askary, S. et al.: Web services business process execution language version 2.0. Technical report, WS-BPEL TC OASIS, January 2007.
- [HuWi2007] Huth, S., Wieland, T.: Geschäftsprozessmodellierung mittels Software-Services auf Basis der EPK. In: Nissen, V., Petsch, M., Schorcht, H. (Hrsg.): Service-orientierte Architekturen. Deutscher Universitätsverlag, Wiesbaden, 2007, S.61-76.
<http://www.c-fis.de/fileadmin/services-EPK.pdf>
- [ODAH2006] Ouyang, C.; Dumas, M.; van der Aalst, W.; ter Hofstede, A.: From business process models to process-oriented software systems: The BPMN to BPEL way. BPM Center Report BPM-06-27, BPM center.org, 2006.
<http://is.tn.tue.nl/staff/wvdaalst/BPMcenter/reports/2006/BPM-06-27.pdf>
- [OADH2006] Ouyang, C.; van der Aalst, W.; Dumas, M.; ter Hofstede, A.: Translating BPMN to BPEL. BPM Center Report BPM-06-02, BPMcenter.org, 2006. <http://is.tn.tue.nl/staff/wvdaalst/BPMcenter/reports/2006/BPM-06-02.pdf>
- [ReMe2006] Recker, J.; Mendling, J.: On the translation between BPMN and BPEL: Conceptual mismatch between process modeling languages. In T. Latour und M. Petit, Hrsg., Proceedings of Workshops and Doctoral Consortium for the 18th International Conference on Advanced Information Systems Engineering. Namur University Press, Luxembourg, 2006.
<http://wi.wu-wien.ac.at/home/mending/publications/06-EMMSAD.pdf>
- [Sche2006] Scheer, A.-W.: ARIS – Business Process Modelling. Springer: New York et al., 3. Auflage, 2006.
- [Whit2004] White, S.A.: Business Process Modeling Notation (BPMN) version 1.0. Business Process Management Initiative, bpm.org, 2004. <http://www.omg.org/docs/bei/05-08-07.pdf>
- [Whit2005] White, S.A.: Using BPMN to model a BPEL process. BPTrends, 3(3):1–18, 2005.