

MUSTERGÜLTIGE PROZESSE – PROZESSMODELLIERUNG UND -AUTOMATISIERUNG AUF BASIS VON WORKFLOW PATTERNS

Patterns (Muster) sind in der Softwareentwicklung verbreitete Lösungsmuster für wiederkehrende Problemstellungen. Diesem Vorbild folgend haben Prof. v. d. Aalst et al. unter dem Titel „Workflow Patterns“ im Jahr 2002 (aktualisierte und erweiterte Fassung im Jahr 2006) eine Reihe von Patterns definiert und veröffentlicht, die häufig wiederkehrende Kontrollflusszenarien in der Prozessmodellierung beschreiben. Der Artikel stellt die Workflow Patterns kurz allgemein vor und zeigt mögliche Anwendungsbeispiele auf. Anschließend wird der Einsatz der Workflow Patterns bei der Transformation fachlicher Geschäftsprozessmodelle in ausführbare Geschäftsprozessmodelle im Rahmen einer prozessbasierten serviceorientierten Architektur (SOA) am Beispiel der Ereignisgesteuerten Prozesskette (EPK) und der Business Process Execution Language (BPEL) erläutert.

Patterns als Lösungsansatz für wiederkehrende Problemstellungen

Patterns sind in der Informatik und insbesondere in der Softwareentwicklung verbreitet als Lösungsansätze für wiederkehrende Anforderungen und Problemstellungen. Ein Beispiel für die Definition eines Patterns in diesem Kontext ist die von S. H. Kaisler: „A design pattern is an abstract solution to a problem“ (vgl. [KAI2005]). Kaisler beschreibt auch die Entstehung eines Designpatterns und vergleicht diese mit dem Prozess des Problemlösens: Jemand findet und dokumentiert die Lösung für ein spezifisches Problem. In vergleichbaren Situationen wird auf diese Lösung zurückgegriffen, und neue Erfahrungen werden hinzugefügt. Dies führt dann zu einem Standardansatz bei der Lösung ähnlicher Probleme und somit zur Definition eines Patterns. Die Weiterentwicklung und Verbesserung des Patterns erfolgt beispielsweise durch die Veröffentlichung oder Weitergabe des Patterns an andere, die ihre Erfahrungen und Erkenntnisse hinzufügen.

Ein wichtiger Meilenstein der Entwicklung von Patterns in der Informatik ist zweifelsohne die Veröffentlichung der objektorientierten Software Patterns durch die „Gang of Four“ um Erich Gamma im Jahr 1995 (vgl. [GAM1995]), der wohl bislang bekanntesten und am weitesten verbreiteten Software Patterns.

Workflow Patterns – Muster für Kontrollflüsse in Prozessen

In ihrem im Jahr 2002 veröffentlichten Dokument mit dem Titel „Workflow Patterns“ erläutern W. M. P. van der Aalst et al. von der technischen Universität Eindhoven regelmäßig im Kontrollfluss von Prozessen auftretende Szenarien (vgl. [AAL2002]). Sie verfolgen dabei nach eigener Aussage die Zielsetzung, systematisch und mit steigender Komplexität die Anforderungen an Prozesse bzw. Workflows zu beschreiben. Die vorgestellten Patterns sind in verschiedene thematische Gruppen aufgeteilt und werden in einer einheitlichen Struktur vorgestellt (vgl. **Tabelle 1**).

Zur Verdeutlichung des Praxisnutzens werden zu jedem Pattern konkrete Beispiele aus der realen (Geschäfts-) Welt genannt; in einigen Fällen sind diese konkreten Anwendungsbeispiele für das Verständnis eines Patterns sogar erforderlich.

Im Jahr 2006 haben die Autoren eine überarbeitete und erweiterte Fassung ihrer Workflow Patterns veröffentlicht, die jetzt insgesamt 43 Patterns umfasst (vgl. [AAL2006]).

Um an dieser Stelle die Idee und Intention der Workflow Patterns etwas konkreter zu machen, sei ein Beispiel aus der Gruppe der „Advanced Branching and Synchronization Patterns“ genannt: Das Pattern „Discriminator“ beschreibt eine Situation in einem Prozess, in der mehrere



Ingo Meier hat Wirtschaftsinformatik studiert und ist als IT Consultant für OPITZ CONSULTING tätig. Er ist Mitglied des OPITZ CONSULTING SOA Competence Center und beschäftigt sich mit den Themen Geschäftsprozessmanagement, Prozessautomatisierung und Serviceorientierte Architekturen.

Aktivitäten parallel gestartet werden. Sobald eine der Aktivitäten beendet ist, wird der Prozessfluss nach den parallel gestarteten Aktivitäten fortgesetzt, die Fertigstellung der übrigen parallel gestarteten Aktivitäten wird ignoriert. Ein praktisches Anwendungsbeispiel für dieses Pattern ist beispielsweise das Korrekturlesen eines Dokumentes: Mehrere Lektoren erhalten gleichzeitig das Dokument zum Lesen, sobald einer von ihnen die Druckreife des Dokumentes meldet, kann der Druck angestoßen werden.

Einsatzmöglichkeiten für die Workflow Patterns

Wie können nun die Workflow Patterns in der Praxis eingesetzt werden? Das wichtigste Anwendungsszenario dürfte wohl die Entwicklung und Dokumentation wiederverwendbarer Lösungen bei der Arbeit mit Prozessen bzw. Workflows sein. Insbesondere bei den komplexeren Patterns vereinfacht und beschleunigt der Rückgriff auf eine bereits vorhandene und valide Lösung die Entwicklung fachlicher und technischer Prozessmodelle.

Ein weiteres Einsatzgebiet der Patterns ist die Evaluierung von Prozessmodellierungs- und Prozessausführungssprachen. Hier existiert eine Vielzahl an Dokumenten und Evaluationen, die die Abbildung der Patterns untersuchen (vgl. [WEB01]) – beispielsweise zu Prozesssprachen wie EPK, BPMN, UML, BPEL. Diese Evaluationen bieten die Möglichkeit,

	EPK	BPEL
Basic Control Patterns		
Sequence	+	+
Parallel Split	+	+
Synchronization	+	+
Exclusive Choice	+	+
Simple Merge	+	+
Advanced Branching and Synchronization Patterns		
Multiple Choice	+/-	+
Synchronizing Merge	+/-	+
Multiple Merge	-	-
Discriminator	-	-
N-out-of-M Join	-	-
Structural Patterns		
Arbitrary Cycles	+	-
Implicit Termination	+	+
Patterns Involving Multiple Instances		
MI without synchronization	-	+
MI with a priori known design time knowledge	-	+
MI with a priori known runtime knowledge	-	-
MI with no a priori known runtime knowledge	-	-
State-based patterns		
Deferred Choice	-	+
Interleaved Parallel Routing	-	+/-
Milestone	-	-
Cancellation Patterns		
Cancel Activity	-	+
Cancel Case	-	+

+ = unterstützt, - = nicht unterstützt, +/- = teilweise unterstützt

Tabelle 1: Direkte Unterstützung der Workflow Patterns in EPK und BPEL

die Kontrollflüsseigenschaften verschiedener Modellierungssprachen zu vergleichen. Darüber hinaus werden auch Laufzeitprodukte für Geschäftsprozesse (Process/Workflow Engines) auf ihre Unterstützung der Patterns hin untersucht (vgl. [WEB02]). Potentielle Kunden formulieren in ihren Fragenkatalogen an Produkthersteller teilweise die Unterstützung ausgewählter Workflow Patterns als Anforderung.

Prozessautomatisierung

Die im vorangegangenen Abschnitt erwähnten Evaluierungen von Prozessmodel-

lierungs- und Prozessausführungssprachen können bei der Automatisierung fachlicher Prozessmodelle – also der Übertragung der Prozesse auf IT-Systeme – eingesetzt werden. Dieser Ansatz soll im Folgenden am Beispiel der Ereignisgesteuerten Prozesskette (EPK) und der Business Process Execution Language (BPEL) erläutert werden.

Ausgangspunkt dieses Szenarios ist eine fachliche Geschäftsprozessmodellierung, die im konkreten Fall auf Basis der ARIS-Methodik (Prof. Scheer) mit EPK-Modellen für die Steuerungs- bzw. Prozesssicht durchgeführt wird. Mögliche Alternativen

wären beispielsweise der Einsatz von BPMN (Business Process Modeling Notation) oder (UML Unified Modeling Language). Diese Modellierung bildet zunächst im Rahmen eines Prozessmanagements die Wertschöpfung im Unternehmen aus einer rein betriebswirtschaftlichen Prozesssicht ab. Die EPK (vgl. Abb. 1) besteht im Wesentlichen aus zwei Objekten: Funktion und Ereignis. Kontrollflussbildende Elemente sind Kanten, die die Objekte miteinander verbinden, und die logischen Operatoren AND, OR und XOR. Diese ermöglichen parallele bzw. alternative Verarbeitung von Pfaden sowie die Synchronisation aufgesplitteter Pfade (vgl. [MEN2005]).

Im nächsten Schritt sollen automatisierbare Teile dieser betriebswirtschaftlichen Prozesse in technische BPEL-Prozesse überführt werden. Ein methodisches Vorgehen sowie eine exemplarische Werkzeugunterstützung für die Überführung von EPK-Prozessmodellen in BPEL-Prozessmodelle werden in [STA 2006] erläutert. Die Methodik verwendet Services als Bausteine, die die fachlichen Funktionen eines Prozesses unterstützen, sodass letztlich eine prozessbasierte Serviceorientierte Architektur (SOA) realisiert wird.

Die Business Process Execution Language (BPEL) ist eine XML-basierte Prozessbeschreibungssprache, die die Automatisierung von Geschäftsprozessen ermöglicht (vgl. [WIN2007]). Seit April

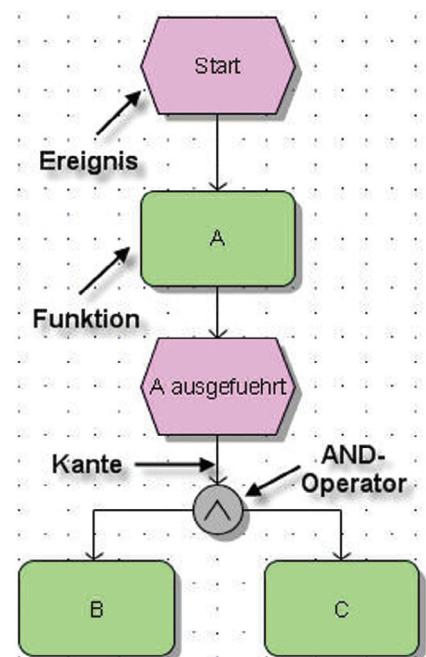


Abbildung 1: Kontrollfluss in einer EPK

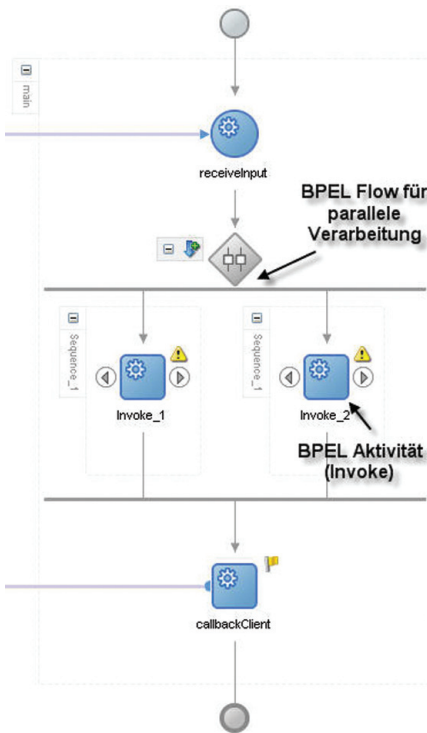


Abbildung 2: Kontrollfluss in einem BPEL Prozess

2007 ist BPEL in der Version WS-BPEL 2.0 ein Standard der OASIS (vgl. (WEB03)), zuvor wurde die Spezifikation von den großen Softwareherstellern (u. a. IBM, Microsoft, SAP, Oracle) entwickelt. Die Funktionen eines fachlichen Prozesses werden in BPEL durch den Aufruf von Web Services implementiert, sodass der BPEL-Prozess im Wesentlichen die Ablauflogik und die Services die Geschäftslogik zur Verfügung stellen. Bei entsprechender Werkzeugunterstützung kann die Modellierung der BPEL Prozesse grafisch erfolgen (vgl. Abb. 2). Die kontrollflussbildenden Elemente in BPEL sind eng an die bekannter Programmiersprachen angelehnt. So existieren beispielsweise While-, Switch- und Flow-Aktivitäten für wiederholte,

alternative oder parallele Verarbeitung.

Einsatz der Workflow Patterns bei der Prozessautomatisierung

Verschiedene Prozessmodellierungssprachen und -standards verfügen über unterschiedliche Modellierungskonstrukte und Notationsmöglichkeiten. Dies bezieht sich sowohl auf die Modelle und deren Eigenschaften, Elemente und Parameter – beispielsweise fachliche und technische Informationen – als auch auf die Abbildung der Kontrollflüsse. Letztere werden hier für die fachliche Modellierung in EPK-Notation und technisch ausführbare Prozessmodelle in BPEL am Beispiel der Workflow Patterns genauer betrachtet. Die zentrale Idee ist, die Abbildung der Workflow Patterns sowohl in EPK-Notation als auch in BPEL zu realisieren. Auf dieser Basis lässt sich dann ermitteln, ob ein EPK-Modell, das einen bestimmten Kontrollfluss enthält, grundsätzlich in einen BPEL Prozess überführt werden kann.

Ein einfaches Beispiel: Das Workflow Pattern „Exclusive Choice“ (vgl. Abb. 3) beschreibt das Kontrollfluss-Szenario, in dem der Kontrollfluss in alternative Pfade aufgesplittet wird. Genau einer dieser alternativen Pfade darf durchlaufen werden. Die anschließende Zusammenführung dieser Pfade wird vom Workflow Pattern „Simple Merge“ (vgl. Abb. 4) beschrieben, bei dem die nachfolgende Aktivität ohne Synchronisation aktiviert wird, sobald einer der eingehenden Pfade navigiert wurde.

Der Kontrollfluss, den diese beiden Patterns beschreiben, kann in einer EPK mit dem XOR-Operator realisiert werden (vgl. Abbildung 5). Das XOR steht dabei

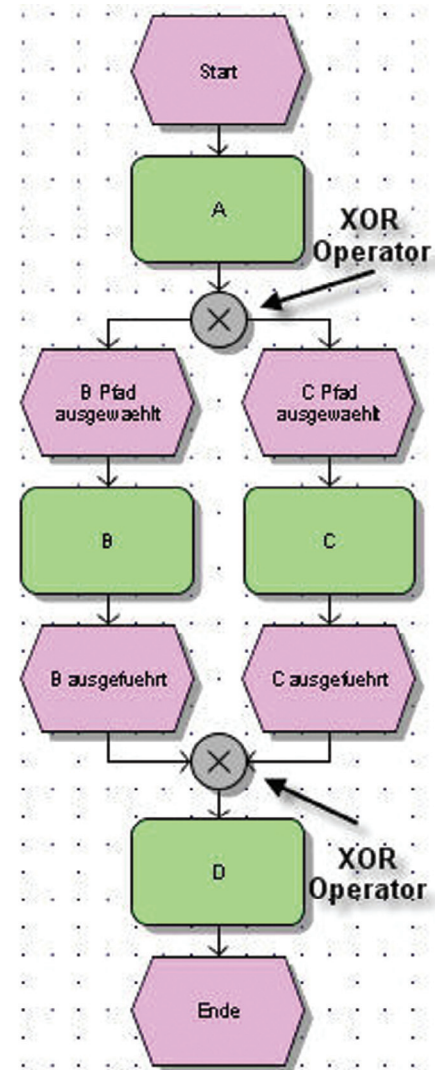


Abbildung 5: Workflow Patterns 'Exclusive Choice' und 'Simple Merge' in EPK-Notation

sowohl an der Stelle, an der die Pfade aufgesplittet werden, als auch bei der Zusammenführung der alternativen Pfade. Somit ist auch die Vorbedingung des „Simple Merge“-Patterns, niemals mehrere Pfade parallel zu navigieren, erfüllt.

Für die Transformation der gerade vorgestellten EPK in einen BPEL Prozess muss – unter anderem – der abgebildete Kontrollfluss in BPEL realisierbar sein. Die beiden Workflow Patterns „Exclusive Choice“ und „Simple Merge“ lassen sich in BPEL mit der Switch-Aktivität abbilden, welche, wie in einigen Programmiersprachen auch, mehrere case-Zweige alternativ verarbeitet (vgl. Abb. 6 und BPEL-Quellcode in Listing 1).

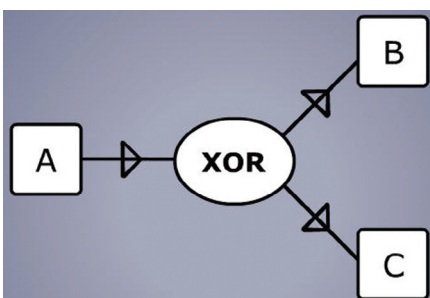


Abbildung 3: Workflow Pattern 4: Exclusive Choice (Quelle: [WEB04])

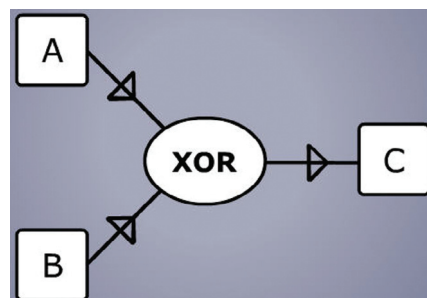


Abbildung 4: Workflow Pattern 5: Simple Merge (Quelle: [WEB04])


```

<sequence>
<receive ..."/>
<invoke name="A" ..."/>
<switch name="XOR-Regel">
<case condition="cond1">
<invoke name="B" ..."/>
</case>
<case condition="cond2">
<invoke name="C" ..."/>
</case>
</switch>
<invoke name="D" ..."/>
<reply name="Ende" ..."/>
</sequence><P>
    
```

Listing 1: BPEL Quellcode zu Workflow Patterns ‚Exclusive Choice‘ und ‚Simple Merge‘ (Ausschnitt)

Für das gerade gezeigte Beispiel existiert nun – im Sinne eines Patterns – eine wieder verwendbare Lösung für beide Prozesssprachen. Wann immer das Pattern „Exclusive Choice“ in einer EPK gefunden wird, kann dieser Kontrollfluss unmittelbar in einen entsprechenden BPEL-Prozess übertragen werden.

Ein weiteres Beispiel: Das Workflow Pattern „Arbitrary Cycles“, bei dem im Sinne einer Schleife Aktivitäten mehrfach durchgeführt werden. Die Besonderheit dieses Patterns ist, dass die Rücksprünge beliebig ineinander geschachtelt sein können. In der EPK-Notation kann das Pattern abgebildet werden, da die EPK ineinander geschachtelte Rücksprünge erlaubt. BPEL hingegen unterstützt nur strukturierte Schleifen, so dass der vom Workflow Pattern „Arbitrary Cycles“ geforderte Kontrollfluss nicht abgebildet werden kann. Folglich handelt es sich hier um ein Beispiel, bei dem in der EPK-Notation vorgenommene Modellierungen nicht nach BPEL übertragen werden können.

Direkt unterstützte Transformationen und mögliche Erweiterungen

Tabelle 1 zeigt eine Übersicht der in EPK Notation und in BPEL abbildbaren Workflow Patterns, abgeleitet aus den bereits vorhandenen Evaluationen der Prozesssprachen (vgl. [WEB01]). Dies bedeutet, dass die Tabelle die direkt von der EPK Notation und vom BPEL Standard unterstützten Patterns auflistet. Dabei wird deutlich, dass die einfachen Patterns in beiden Notationen direkt unterstützt werden und dementsprechend leicht abgebildet werden können. Bei den komplexeren Kontrollflüssen hingegen ist eine direkte

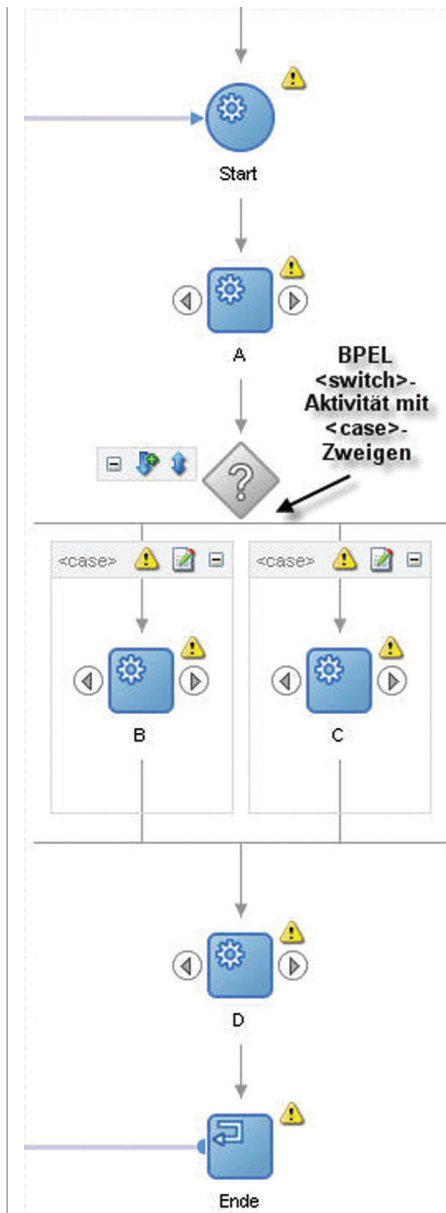


Abbildung 6: Workflow Patterns ‚Exclusive Choice‘ und ‚Simple Merge‘ in BPEL (Oracle JDeveloper)

Unterstützung der Patterns häufig nicht gegeben.

Für die in beiden Prozesssprachen unterstützten Kontrollflüsse lässt sich die Transformation zwischen EPK und BPEL auch automatisieren. Ein Verfahren, das beispielsweise der IDS Scheer ARIS SOA Architect oder die Oracle BPA Suite anwenden. Dabei wird im Kontrollfluss der EPK nach Workflow Patterns gesucht und anschließend die Transformation nach BPEL durchgeführt. Diese Produkte ermöglichen das Generieren von BPEL-Prozessartefakten direkt aus EPK Modellen.

Literaturliste

[AAL2002] van der Aalst, W. M. P. et al.: Workflow Patterns, Technical report, 2002, <http://is.tm.tue.nl/research/patterns/download/wfs-pat-2002.pdf>

[AAL2006] van der Aalst, W. M. P. et al.: Workflow Control-Flow Patterns: A Revised View., 2006, <http://www.workflowpatterns.com/documentation/documents/BPM-06-22.pdf>

[GAM1995] Gamma, E. et al.: Design Patterns: Elements of Reusable Object-Oriented Software, 1995

[KAI2005] KAISLER, S. H.: Software Paradigms, 2005

[MEN2005] Mendling, Jan et al.: Towards Workflow Pattern Support of Event-Driven Process Chains (EPC), 2005, <http://wi.wu-wien.ac.at/home/mendling/XML4BPM2005/xml4bpm-2005-proceedings-mendling.pdf>

[STA2006] Stähler, Dirk: Oracle verbindet Fach- und IT-Modell, 2006, <http://www.computerwoche.de/heftarchiv/2006/42/1216430/>

[WIN2007] Winterberg, Torsten: BPEL wird erwachsen, 2007, http://www.opitz-consulting.de/pdf/veroeffentlichungen/javamagazin_07_07.pdf

[WEB01] <http://is.tm.tue.nl/research/patterns/standards.htm>

[WEB02] <http://is.tm.tue.nl/research/patterns/products.htm>

[WEB03] www.oasis-open.org/committees/wsbpel

[WEB04] <http://is.tm.tue.nl/research/patterns>

Homepage Workflow Patterns unter <http://www.workflowpatterns.com>

Um die Unterstützung weiterer Workflow Patterns in der EPK-Notation zu ermöglichen, schlagen Mendling et al. in ihrem Whitepaper Erweiterungen der EPK-Notation vor (vgl. [MEN2005]).

In aktuell am Markt verfügbaren BPEL-Laufzeitumgebungen (z.B. Oracle BPEL Process Manager, IBM WebSphere Process Server) können in der Regel mehr Workflow Patterns umgesetzt werden, als direkt vom BPEL-Standard unterstützt werden. Dies lässt sich beispielsweise durch die Kombination mehrerer BPEL-Aktivitäten, durch Sub-Prozesse oder durch



Events und Event Handler realisieren. Die Unterstützung der Patterns erfolgt dann indirekt, gegebenenfalls sogar produktabhängig (vgl. [WEB02]). Auch für diese indirekten Szenarien bietet es sich an, nach wieder verwendbaren Lösungen bzw. Mustern zu suchen.

Abschließend noch ein Hinweis zur Umsetzung der Workflow Patterns in BPEL: Die in der Tabelle sowie im Text

referenzierten Transformationen beziehen sich auf die Version 1.1 der BPEL Spezifikation (BPEL4WS 1.1). Mit dem im April dieses Jahres verabschiedeten BPEL 2.0 Standard wurden dem Sprachumfang von BPEL neue Aktivitäten und Konstrukte hinzugefügt, welche teilweise die direkte Unterstützung weiterer Workflow Patterns erlauben.

Fazit

Die Transformation von Kontrollflüssen in Prozessen stellt eine wesentliche Anforderung bei der Automatisierung fachlicher Geschäftsprozesse dar. Im Artikel wurde am Beispiel EPK und BPEL gezeigt, wie eine solche Transformation systematisch auf der Basis der Workflow Patterns von v. d. Aalst et al. durchgeführt werden kann und welchen Nutzen dabei ein patternbasierter Ansatz im Hinblick auf wieder verwendbare Lösungen bietet.