

Anforderungen an die Modellierung und Ausführung von Prozessvarianten

Alena Hallerbach¹, Thomas Bauer¹ und Manfred Reichert^{2,3}

¹ Daimler AG - Group Research & Adv. Engineering, GR/EPD, Ulm, Deutschland,
{alena.hallerbach; thomas.tb.bauer}@daimler.com

² Information Systems Group, Universität Twente, Niederlande,
m.u.reichert@cs.utwente.nl

³ Institut f. Datenbanken und Informationssysteme, Universität Ulm, Deutschland,
manfred.reichert@uni-ulm.de

Zusammenfassung. Die Modellierung von Geschäftsprozessen sowie deren Ausführung mittels Workflow-Management-Systemen (WfMS) bilden zentrale Aufgaben bei der Realisierung prozessorientierter Informationssysteme. In der Praxis hat sich gezeigt, dass ein Prozesstyp oftmals in zahlreichen Varianten auftritt, die Anpassungen an bestimmte Rahmenbedingungen darstellen. Die adäquate Modellierung und Ausführung solcher Prozessvarianten stellt eine große Herausforderung dar, der heutige Geschäftsprozessmodellierungswerkzeuge und WfMS bei weitem nicht gerecht werden. Diese ermöglichen lediglich das Ausmodellieren aller Prozessvarianten in separaten Prozessmodellen, wodurch ein hoher Wartungsaufwand resultiert: Werden Prozessanpassungen erforderlich, sind meist mehrere Varianten betroffen und deshalb mehrere Prozessmodelle anzupassen. Dies wiederum führt schnell zu Inkonsistenzen. Um dies zu vermeiden, sollten Varianten in *einem* Prozessmodell modelliert und explizit als Varianten gekennzeichnet werden. Weitergehend sollte während der Ausführung solcher variantenbehafteter Prozesse flexibel auf veränderte Rahmenbedingungen reagiert werden können, z.B. indem dynamisch zwischen Prozessvarianten gewechselt wird. Dieser Beitrag untersucht Anforderungen an die Modellierung und Ausführung von variantenbehafteten Prozessen und skizziert Lösungsideen.

1 Motivation

Die Arbeitsabläufe in Unternehmen werden oftmals in Prozessmodellen dokumentiert, für deren Erstellung Werkzeuge wie ARIS Business Architect [IDS06], ADONIS [BOC07] oder WBI Modeler [IBM07] zum Einsatz kommen. Ein einzelnes Prozessmodell beschreibt üblicherweise einen bestimmten Ablauftyp (z.B. Kreditantragstellung, Reisekostenabrechnung) und legt u.a. die durchzuführen- den Aktivitäten, deren Ausführungsreihenfolge sowie die zu ihrer Ausführung erforderlichen Ressourcen fest. Mit der expliziten Modellierung dieser Aspekte werden verschiedene Ziele verfolgt: Durch die Dokumentation der Arbeitsabläufe in Geschäftsprozessmodellen werden diese für die Prozessbeteiligten transparenter und können z.B. durch Analyse und Simulation verbessert werden

[Sch98,Gad05,Wes07]. Eine andere Anwendung stellt die Automatisierung von (Teil-) Prozessen dar, z.B. unterstützt durch den Einsatz eines Workflow-Management-Systems (WfMS). In diesem Fall wird ein Workflow-Modell erstellt, das zur Ausführung in einem WfMS instanziiert wird. Das WfMS steuert dann die Ausführung der einzelnen Prozessaktivitäten entsprechend der modellierten Reihenfolge und weist sie den Mitarbeitern, unter Bereitstellung der erforderlichen Daten und Anwendungen, zu [DvdAtH05,LR99,GHS95].

Sowohl für Geschäftsprozesse als auch für Workflows werden in der Praxis häufig unterschiedliche, parallel existierende Varianten eines Prozesstyps definiert. Bei der Produktentwicklung etwa entstehen Prozessvarianten abhängig vom zu entwickelnden Produkt (z.B. PKW und LKW) sowie von organisatorischen Zuständigkeiten, strategischen Ausrichtungen und sonstigen Rahmenbedingungen. Ein konkretes Beispiel aus der Produktentwicklung zeigt der in Abbildung 1 dargestellte, stark vereinfachte Prozess zur Abwicklung von Änderungsanträgen: Nachdem ein Änderungsantrag angestoßen wurde, gibt der Prozessverantwortliche die Stellungnahme verschiedener Bereiche in Auftrag. Sind diese Stellungnahmen nicht konfliktfrei, werden sie vom Projektleiter zu einer Gesamtstellungnahme zusammengefasst. Der Änderungskreis (Entscheidungsgremium für Änderungen) entscheidet dann über die Genehmigung oder Ablehnung des Änderungsantrags. Abschließend erfolgt die Umsetzung der beantragten Änderung.

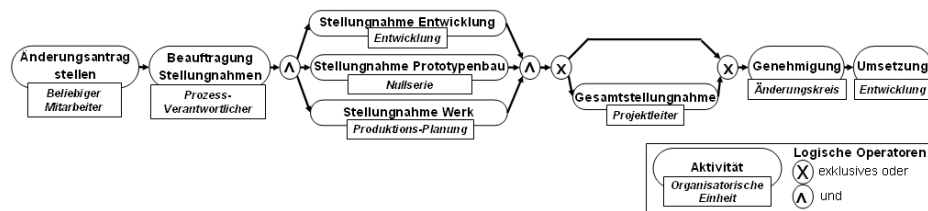


Abb. 1. Vereinfachter Prozess zur Abwicklung eines Änderungsantrags

In der Praxis wird dieser Prozess unter verschiedenen Rahmenbedingungen ausgeführt, die eine Abweichung vom hier dargestellten Standardablauf erforderlich machen. Dies wiederum führt zur Entstehung von Prozessvarianten. In Abbildung 2 sind drei mögliche Prozessvarianten unseres Beispielprozesses aufgeführt: Prozessvariante 1 kommt der Forderung nach höheren Qualitätsansprüchen durch Einholen einer zusätzlichen Stellungnahme der Abteilung für Qualitätssicherung entgegen. Prozessvariante 2 wird während der Serienfertigung des Fahrzeugmodells relevant, wenn zusätzlich der Einkauf eine Stellungnahme bezüglich der zu erwartenden Teilekosten abgeben soll. Durch Prozessvariante 3 wird die Bearbeitung der Stellungnahmen auf Grund hohen Zeitdrucks beschleunigt, indem die Verantwortung nicht auf viele verschiedene Abteilungen verteilt, sondern von der Entwicklung selbst getragen wird. Diese informiert den Einkauf dann nur noch über die zu erwartenden Kosten der Änderung.

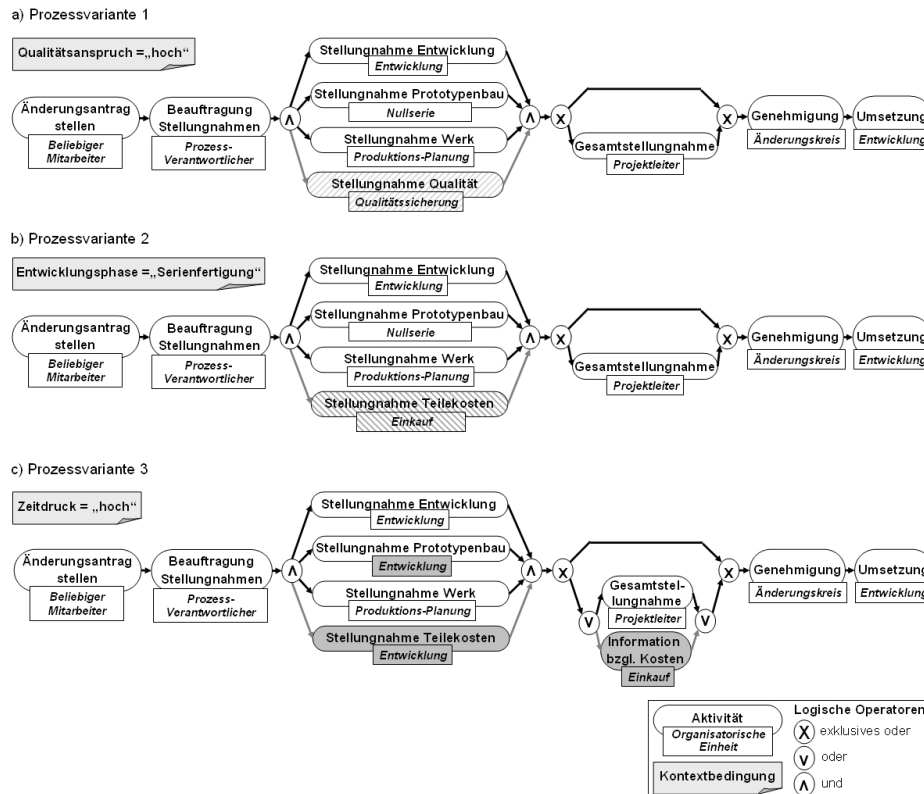


Abb. 2. Varianten des Prozesses zur Abwicklung eines Änderungsantrags

Obwohl die Modellierung solcher Prozessvarianten eine hohe Praxisrelevanz hat, bieten existierende Prozessmodellierungswerkzeuge keine adäquate Unterstützung. Sie erfordern das Ausmodellieren jeder einzelnen Prozessvariante, wie in Abbildung 2 dargestellt. Sind für einen Prozess viele Varianten zu modellieren, gelangt der Modellierer schnell an den Punkt, an dem die Vielzahl der Prozessmodelle für ihn nicht mehr handhabbar bzw. wartbar ist. Kosten und Aufwände für die Wartung der Prozessmodelle steigen, da bei einer Änderung im Prinzip jedes Prozessmodell separat angepasst werden muss. Damit einher geht eine hohe Fehleranfälligkeit, die zu inkonsistenten Prozessmodellen führt.

Die Herausforderung besteht nun darin, Kosten und Aufwände für die Modellierung und Wartung von Prozessvarianten zu reduzieren. Dies soll durch die Modellierung der Prozessvarianten in einem Prozessmodell erreicht werden.

Bei der Entwicklung eines Lösungsansatzes sind auch spätere Phasen des Prozesslebenszyklus [vdAHW03, WWRD07] zu berücksichtigen, um z.B. das dynamische Wechseln zwischen zwei Prozessvarianten auch noch während der Ausführung zu ermöglichen, wenn dies eine Veränderung der Rahmenbedingungen erforderlich macht.

Basierend auf einer im Automobilbereich durchgeführten Fallstudie, stellt der vorliegende Beitrag grundlegende Anforderungen an die Modellierung und Verwendung von Prozessvarianten vor und diskutiert erste Lösungsansätze. Kapitel 2 beschreibt die grundlegende Idee einer integrierten Verwaltung von Prozessvarianten und dient der Begriffsklärung. Anschließend werden in Kapitel 3 die Anforderungen an die Modellierung bzw. Darstellung von Prozessvarianten beschrieben. Aspekte der Ausführung von Prozessvarianten in einem WfMS und damit verknüpfte Anforderungen sind Gegenstand von Kapitel 4. Der Beitrag schließt mit einer Zusammenfassung über die behandelten Themen und gibt einen Ausblick auf zukünftige Arbeiten.

2 Genereller Ansatz und Begriffsklärung

Prozessvarianten werden meist auf Basis eines vorhandenen Prozessmodells erstellt und weisen daher eine Ähnlichkeit zu diesem *Basisprozess* auf. Zum Beispiel sind die Prozessvarianten in Abbildung 2 Anpassungen des Prozesses aus Abbildung 1 an veränderte Rahmenbedingungen. Eine Möglichkeit, diese variantenspezifischen Anpassungen des Basisprozesses in einem Modell zu repräsentieren, besteht in der Modellierung der varianten Abläufe mittels bedingter Verzweigungen. Im Prozessmodell aus Abbildung 3 etwa realisiert jeder Pfad eine Prozessvariante. Mit Hilfe der Verzweigungsbedingungen wird zudem eine Ausführung in Abhängigkeit von den gegebenen Rahmenbedingungen erreicht.

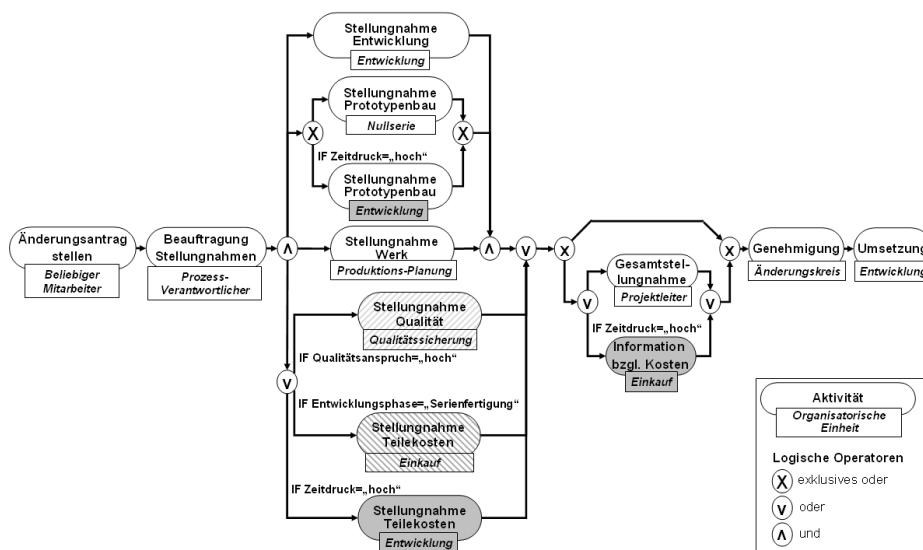


Abb. 3. Modellierung von Prozessvarianten mittels bedingter Verzweigungen

Ein Nachteil dieser naheliegenden Art der Modellierung ist, dass die Prozessmo-

delle sehr groß werden, da die kombinatorische Vielfalt aller möglichen Prozessvarianten abgebildet werden muss. Die Information darüber, wann eine bestimmte Variante durchlaufen wird, ist in der Ablauflogik des Prozessmodells versteckt. Die Varianten sind dann nicht mehr transparent, da sie nicht explizit als solche gekennzeichnet sind. Der Modellierer kann nicht mehr erkennen, ob eine bedingte Verzweigung eine Prozessvariante darstellt oder eine „normale“ Verzweigung im Prozessmodell. Aber nicht nur dem Anwender geht das Modellierungswissen über Varianten verloren, sondern auch dem System. Dies führt dazu, dass es bei der Darstellung des Prozessmodells z.B. nicht möglich ist, nur einzelne Prozessvarianten zu betrachten. Weiter ist es nicht möglich, Varianten separat zu instanziiieren, sondern es muss immer eine Instanz des gesamten Prozessmodells erzeugt werden.

Unser Ansatz soll es erlauben, die Informationen über Prozessvarianten und deren Abhängigkeit von bestimmten Rahmenbedingungen im gesamten Prozesslebenszyklus explizit und transparent zu machen [HBR07]. Wir gehen dazu ebenfalls von einem Basisprozess aus, der zur Bildung der Prozessvarianten angepasst wird. Die Anpassungen des Basisprozesses sind jedoch eigenständige Objekte mit eigener Semantik. Für diese Anpassungen kann explizit definiert werden, unter welchen Rahmenbedingungen sie am Basisprozess vorgenommen werden sollen. In unserem Ansatz verwenden wir dazu *Kontextbedingungen*. Diese definieren mit Hilfe von *Kontextvariablen*, festgelegten Wertebereichen und Vergleichsoperatoren, wann eine Anpassung angewendet werden muss. Der Anwender ist dann in der Lage, durch die Angabe aktueller Werte für ausgewählte Kontextvariablen, die Auswertung der Kontextbedingungen und somit die automatische Konfiguration des Prozessmodells bzw. das Generieren einer spezifischen Prozessvariante anzustoßen. Die Anpassungen werden somit optional definiert, da ihre Anwendung vom aktuellen (Prozess-)Kontext abhängt. Im Folgenden werden sie daher als *Optionen* bezeichnet.

Bei der Modellierung von Optionen muss festgelegt werden, wo die Anpassungen am Basisprozess vorgenommen werden. Diese Positionen werden durch *Aufsetzpunkte* definiert. Jeder Ein- und Ausgang eines Knotens des Basisprozesses kann als Aufsetzpunkt für eine Anpassung verwendet werden. Die Anpassung des Basisprozesses kann dann auf unterschiedliche Weise erfolgen. Die häufigste Maßnahme ist das Hinzufügen von Prozessfragmenten zum Basisprozess. Weitere Änderungsoperationen sind zum Beispiel das Ändern von Attributwerten sowie das Verschieben und Entfernen von Prozessfragmenten [HBR08]. (Für eine allgemeine Betrachtung zu Änderungsmustern in prozessorientierten Anwendungen siehe [WRR07].)

Abbildung 4 zeigt die Grundidee unseres Ansatzes. Im Basisprozess markieren die Aufsetzpunkte a-f, wo Anpassungen vorgenommen werden können. Die modellierten Optionen geben an, welche Änderungen vorgenommen werden sollen (Einfügen, Ändern, etc.) und welche Kontextbedingung zutreffen muss. Das Ergebnis der Anwendung der einzelnen Optionen sind die Prozessmodelle der Varianten wie in Abbildung 2 gezeigt.

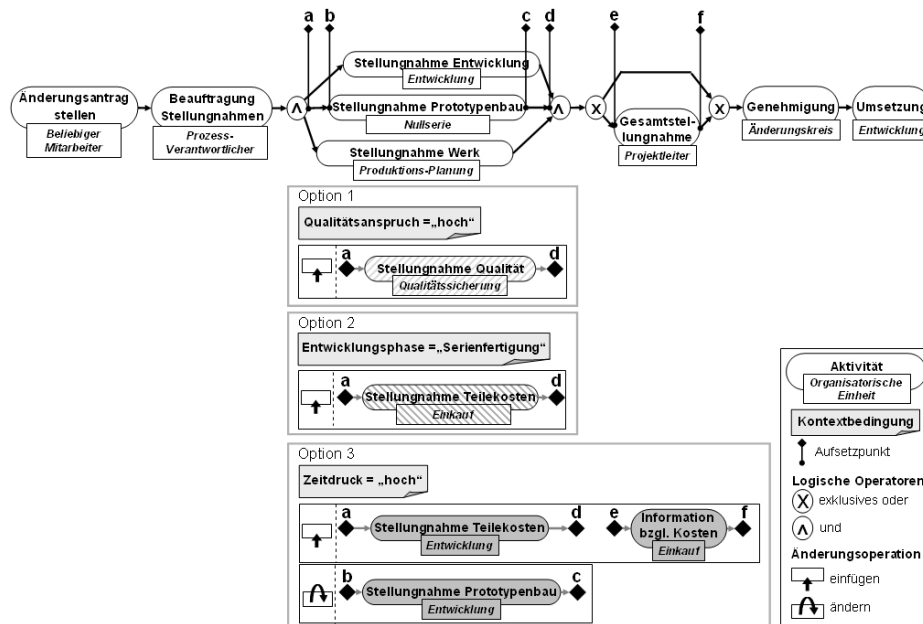


Abb. 4. Definition von Optionen mittels Aufsetzpunkten

3 Modellierung von Optionen

In diesem Kapitel diskutieren wir grundlegende Anforderungen an die Modellierung von Optionen. Zunächst beschreiben wir, wie Optionen modelliert werden sollen (Modellierungsmethode). Anschließend stellen wir Alternativen für die Darstellung von Optionen vor und gehen auf die Verwendung von Kontexten ein. Schließlich beschreiben wir mögliche Konsistenzprobleme bei der Modellierung von Optionen und daraus resultierende Anforderungen.

3.1 Modellierungsmethode

Es soll für den Modellierer intuitiv und mit wenig Aufwand möglich sein, Optionen in seiner Modellierungsumgebung zu erstellen. Grundsätzlich gibt es zur Modellierung von Optionen zwei verschiedene Ansätze:

- **Ansatz 1:** Heutige Prozessmodellierungswerkzeuge erlauben das „Zeichnen“ eines Prozessmodells. Dazu werden Knoten und Kanten verschiedenen Typs z.B. einer Palette entnommen und in der Modellierungsumgebung zu einem Prozessmodell zusammengesetzt. Für die Modellierung von Optionen müssten diese Paletten um spezielle Objekte erweitert werden. So muss für Aufsetzpunkte (vgl. Abbildung 4), Kontextbedingungen und den Typ einer Anpassung (z.B. Einfügen, Ändern, etc.) eine geeignete Repräsentation gefunden werden. Außerdem wird eine Gruppierungsfunktion erforderlich, um Anpassungen in einer Optionen zusammenfassen zu können.

- **Ansatz 2:** Ein anderer Ansatz ist die Modellierung mittels Änderungsoperationen. Der Modellierer ist dabei an Operationen gebunden, die ihm vom Modellierungswerkzeug vorgegeben werden. Solche Änderungsoperationen müßten dann für alle notwendigen Aktionen, z.B. das Einfügen und Bearbeiten von Optionen, zur Verfügung stehen.

Anders als beim Zeichnen des Prozessmodells ist der Modellierer bei Änderungsoperationen an die vom Modellierungswerkzeug vorgegebenen Operationen gebunden und wird daher in seiner Modellierungsfreiheit eingeschränkt. Die Verwendung von Änderungsoperationen bietet allerdings den Vorteil, dass die strukturelle Konsistenz (z.B. keine UND-Synchronisation nach einer ODER-Verzweigung) des Prozesses besser gewährleistet werden kann, wohingegen beim Zeichnen z.B. einzelne Kanten vergessen werden können. Die Änderungen können zudem gut in einer Art Änderungshistorie gespeichert und jederzeit von Benutzer und System nachvollzogen werden.

Die letztliche Festlegung auf einen Ansatz orientiert sich stark daran, ob ein existierendes Modellierungswerkzeug um das von uns beschriebene Variantenmanagementkonzept erweitert oder eine Neuentwicklung angestrebt wird.

3.2 Darstellung von Optionen

Ein Prozessmodell sollte auch bei einer Vielzahl von Optionen übersichtlich und verständlich bleiben. Eine gute Darstellung hilft dem Modellierer z.B. bei der Fehlererkennung sowie bei der Analyse der Modelle. Im Folgenden werden zwei Darstellungsalternativen für Optionen erörtert. Sie dienen als Basis weiterer Überlegungen.

Vollständige Anzeige der Optionen: Das Darstellungsbeispiel aus Abbildung 5a zeigt ein Prozessmodell mit zwei Optionen, die vollständig angezeigt werden. In dieser Darstellung können die Optionen gut miteinander verglichen werden. Die Darstellung wird bei einer größeren Anzahl jedoch unübersichtlich.

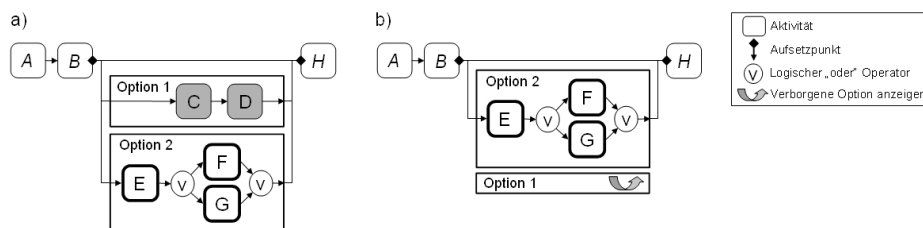


Abb. 5. Vollständige a) und ausgewählte b) Anzeige von Optionen

Auswahl der angezeigten Option: In Abbildung 5b wird lediglich eine Option angezeigt. Andere Optionen können mit Hilfe des Pfeils expandiert werden. Durch diese Darstellung wird die Übersichtlichkeit des Prozessmodells erhöht, die modellierten Optionen sind jedoch schwerer vergleichbar.

Je nach Anwendungsfall ist eine der Darstellungsalternativen der jeweils anderen vorzuziehen. Daher müssen beide Alternativen vom System unterstützt werden. Um die Vorteile beider Darstellungen in vollem Umfang nutzen zu können, muss ein automatisch erzeugbarer Wechsel zwischen den Alternativen möglich sein. Eine zusätzliche Anforderung ist dann, dass beim Wechsel Usability-Aspekte berücksichtigt werden, wie z.B. das Beibehalten der Position der Elemente auf dem Bildschirm (vgl. „Mental Map“ [MELS95]).

3.3 Kontext

Für die Modellierung von Optionen muss festgelegt werden, unter welchen Bedingungen die Optionen zur Anwendung kommen sollen. In unserem Ansatz werden zu diesem Zweck Kontextbedingungen formuliert, auf deren Grundlage eine Konfiguration des Prozessmodells möglich wird. Zur Beschreibung der Kontextbedingungen können beliebige Kontextvariablen aus dem Prozessumfeld definiert werden.

Es ist zu beobachten, dass Kontextvariablen oftmals in semantischer Beziehung zu einander stehen. Solche Beziehungen können z.B. sein:

1. Gültige Kontextkombination:
WENN Baureihe = „B-Klasse“
DANN Geschäftsbereich = „Mercedes“
2. Ungültige Kontextkombination:
WENN Baureihe = „B-Klasse“
DANN Geschäftsbereich = „Daimler Financial Services“

Das zweite Beispiel stellt keine sinnvolle Kombination der Kontextvariablen bzw. ihrer Werte dar, da der Geschäftsbereich „Daimler Financial Services“ keine Fahrzeuge entwickelt; diese Beziehung kann daher als ungültig bewertet werden.

Nur für gültige Kombinationen von Kontextvariablen sollen Kontextbedingungen definierbar sein, die dann Optionen zugewiesen werden können. Die Kontextbeziehungen sind daher entsprechend zu erfassen und im System abzubilden. Im Folgenden werden die Konzepte *Kontextwürfel*, *Kontextgraph* und *Kontextregel* zur Repräsentation der Beziehungen von Kontextvariablen vorgestellt:

Kontextwürfel: Gültige bzw. ungültige Kombinationen von Kontextvariablen bzw. deren Werten können in einem n-dimensionalen Würfel definiert werden. Abbildung 6 zeigt ein Beispiel: An den Achsen sind die Werte der Kontextvariablen *Baureihe*, *Geschäftsbereich* und *Entwicklungsphase* aufgetragen. Die Notation ist in unserem Beispiel so gewählt, dass gültige Kombinationen weiß und ungültige Kombinationen grau dargestellt sind.

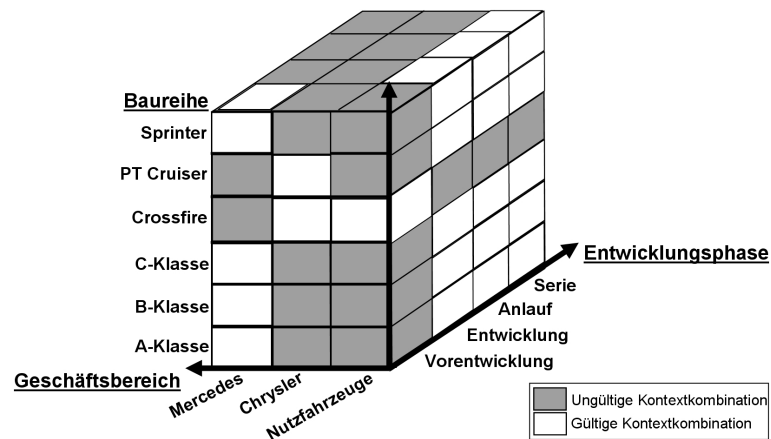


Abb. 6. Kontextwürfel mit gültigen (weiß) und ungültigen (grau) Kontextkombinationen

Unter der Annahme, dass alle n Kontextvariablen m mögliche Werte annehmen können, gibt es m^n mögliche Kontextkombinationen. Für 3 Kontextvariablen mit je 3 Werten ergibt dies 27 Kontextkombinationen, bereits für 7 Kontextvariablen mit je 7 Werten gibt es über 800000 Kontextkombinationen. Das Zahlenbeispiel zeigt, dass bereits bei einer geringen Anzahl an Kontextvariablen und Werten das Bewerten aller Kontextkombinationen hinsichtlich Gültigkeit sehr aufwendig wird. Außerdem wird die Darstellung eines n -dimensionalen Würfels unübersichtlich und komplex. Es wird daher eine Lösung benötigt, die ein benutzerfreundliches Erstellen und Verwalten einer großen Zahl an Kontextvariablen und ihrer Werte erlaubt.

Kontextgraph: Ein alternativer Ansatz sind gerichtete Kontextgraphen. Abbildung 7a zeigt die Kontextvariablen und Abbildung 7b die zugehörigen Wertebereiche. Die gültigen Kontextkombinationen werden durch die Pfeile zwischen den Kontextvariablen bzw. deren Werte abgebildet. Konkret heißt das, dass der Wert am Beginn des Pfeils einen der Werte am Ende erzwingt. Kombinationen, die nicht mit einem Pfeil verbunden sind (z.B. „Mercedes“ und „PT-Cruiser“), sind ungültig. Diese statische Struktur ist jedoch nur für wenig veränderliche

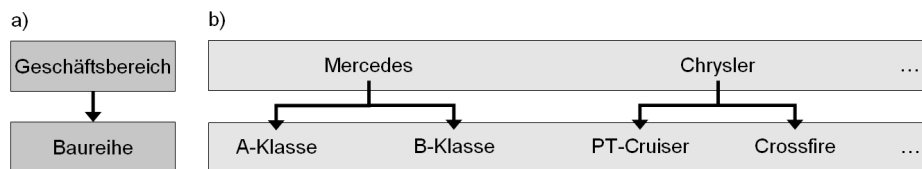


Abb. 7. Kontextvariablen in a) mit Werten in b) und Abhängigkeiten (Pfeile)

Beziehungen sinnvoll nutzbar, da ansonsten der Aufwand zur Wartung und Aktualisierung des Kontextgraphen zu hoch wird.

Kontextregel: Kontextwürfel und Kontextgraphen sind nicht geeignet, um komplexere Beziehungen zwischen Kontextvariablen bzw. deren Werten auszudrücken. Ein mächtigeres Beschreibungsmittel stellen Kontextregeln dar, mit denen sich auch komplexe Abhängigkeiten formal ausdrücken lassen. Beispiel:
WENN Qualitätsanspruch IN {hoch, sehr_hoch} DANN Zeitdruck = „hoch“

Kontextregeln können gut als Ergänzung zu einer graphischen Repräsentation der Beziehungen verwendet werden. Aufgrund der Schwächen des Kontextwürfels verfolgen wir in unserem Ansatz die kombinierte Verwendung von Kontextgraphen und -regeln. Abbildung 8 zeigt, wie die im Kontextgraphen durch Pfeile dargestellten Abhängigkeiten mit Hilfe von Kontextregeln spezifiziert werden können.

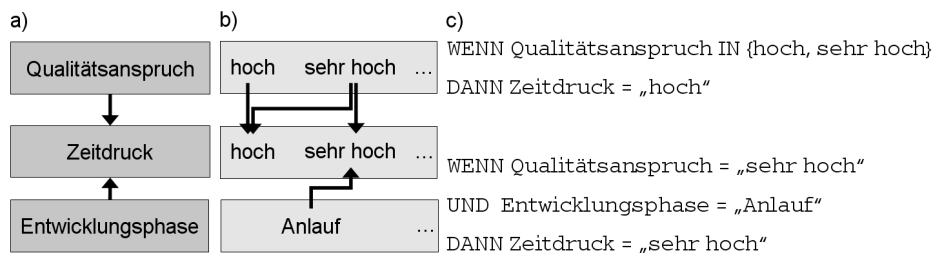


Abb. 8. Spezifizierung von Abhängigkeiten (dargestellt in einem Kontextgraphen in a+b) durch Kontextregeln (c)

Sind die Beziehungen zwischen Kontextvariablen bzw. deren Werten vollständig beschrieben, können darauf aufbauend Kontextbedingungen für Optionen definiert werden. Bei der Auswertung dieser Kontextbedingungen ergibt sich eine weitere interessante Problemstellung: In unserem Beispiel aus Abbildung 2, die ausmodellerte Prozessvarianten darstellt, sind den Prozessvarianten unterschiedliche Kontextbedingungen zugewiesen. Diese sind so gewählt, dass sie sich nicht gegenseitig ausschließen. Das heißt, für einen gegebenen Kontext können mehrere Prozessvarianten gleichzeitig gültig sein. Dies wäre z.B. für den aktuellen Kontext:

Entwicklungsphase = „Serienfertigung“
Zeitdruck = „hoch“

der Fall. Mit heutigen Prozessmodellierungswerkzeugen müßte für diesen Kontext eine zusätzliche Prozessvariante ausmodelliert werden, welche die Kontextbedingungen und die Anpassungen beider Prozessvarianten zusammenführt. Unser Ansatz erfordert dagegen nicht notwendigerweise die Modellierung einer neuen Option. Die Gültigkeit mehrerer Optionen im aktuellen Kontext kann auch

durch eine gleichzeitige Anwendung der Optionen, d.h. einer Kombination der Anpassungen, realisiert werden.

3.4 Konsistenz

Bei der Modellierung muss ein Prozessmodell erzeugt werden, das den Geschäftsprozess vollständig und korrekt beschreibt. Nur dann macht die Dokumentation Sinn bzw. kann das Modell später in ein Workflow-Modell überführt werden. Dazu müssen gewisse Anforderungen an die Konsistenz und Korrektheit des Prozessmodells erfüllt werden, etwa die korrekte Versorgung der Prozessaktivitäten mit Daten oder die korrekte Terminierung des Workflows (einschließlich Verklemmungsfreiheit) [KR96,vdA97].

Die Konsistenz- und Korrektheitsbedingungen müssen auch bei der Modellierung und Verwendung von Optionen berücksichtigt werden. So darf eine Option, die gemäß ihrer Kontextbedingung im aktuellen Kontext gültig ist, nur dann auf den Basisprozess angewendet werden, wenn dies nicht zu einem inkonsistenten bzw. inkorrekten Prozessmodell führt. Auch bei Änderungen am Basisprozess selbst muss die Konsistenz und Korrektheit der Optionen erhalten bleiben, ggf. auch durch entsprechende Anpassung der Optionen (z.B. Aktualisierung der Aufsetzpunkte). Herausforderung ist nun, problematische Prozessmodelle zu erkennen und darauf zu reagieren, z.B. durch Warnung des Modellierers oder Einschränkung der Modellierungsfreiheit. Im Folgenden stellen wir Beispiele für inkonsistente Prozessmodelle vor.

Inkorrekte Datenversorgung: Der Datenfluss eines Prozessmodells beschreibt die Verwendung und Weitergabe von Daten zwischen den Aktivitäten des Prozesses. Zur Verdeutlichung des Problems inkonsistenter Datenflüsse bei der Modellierung von Optionen wird der Datenfluss in unserem Beispiel aus Abbildung 9 explizit modelliert. Option 1 ist im Prozessmodell hinsichtlich des Datenflusses konsistent. Im Knoten C wird ein Datenobjekt geschrieben, dass im Knoten E des Basisprozesses gelesen wird. Ohne Option 1 ist der Basisprozess jedoch nicht konsistent, da die Schreiboperation auf dem Datenobjekt gegebenenfalls fehlt und so der Knoten E nicht korrekt mit Daten versorgt wird.

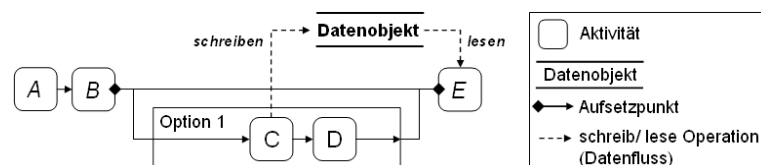


Abb. 9. Inkonsistenter Datenfluss bei Verwendung von Option 2

Verklemmung: Der Kontrollfluss beschreibt die Ausführungsreihenfolge der Aktivitäten eines Prozessmodells. Abbildung 10a zeigt ein Beispiel für einen inkonsistenten Kontrollfluss. Hier ist ein Prozessmodell mit zwei Optionen dargestellt. Im vorangegangenen Abschnitt haben wir beschrieben, dass unter der Annahme, dass ein Kontext definiert wird, für den die Kontextbedingungen mehrerer Optionen zutreffen, die Anpassungen der Optionen kombiniert werden müssen. Im Beispiel aus Abbildung 10a gilt dies für Option 1 und Option 2. Beide Optionen bestimmen, dass die Knoten C und D in den Basisprozess eingefügt werden sollen. Dabei legt Option 1 die Reihenfolge der Knoten so fest, dass C vor D ausgeführt werden soll. In Option 2 wird die umgekehrte Reihenfolge der Knoten definiert. Die Kombination der Anpassungen soll nicht zu einem doppelten Einfügen der Aktivitäten führen, sondern die beschriebenen Kontrollflussabhängigkeiten übernehmen. Das Prozessmodell besitzt dann durch den Pfad C-D-C einen Zyklus und kann nicht terminieren (siehe Abbildung 10b). Der Kontrollfluss ist somit nicht konsistent.

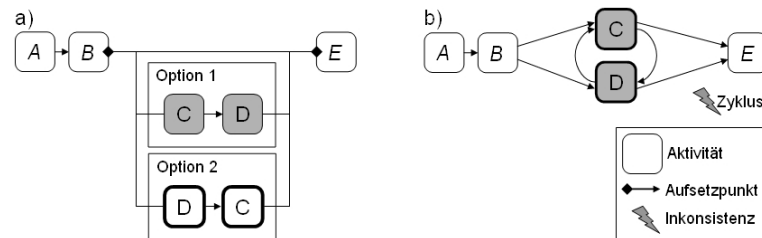


Abb. 10. Inkonsistenter Kontrollfluss durch Kombination von Optionen

Wie das Beispiel aus Abbildung 10 zeigt, ist die konsistente und korrekte Kombination von Optionen nicht immer möglich, da Optionen widersprüchliche Veränderungen am Basisprozess vornehmen können (z.B. setzen unterschiedlicher Attributwerte für eine Aktivität im Basisprozess). Solche Konflikte müssen einerseits vom System erkannt werden und andererseits weitestgehend automatisch aufgelöst werden, d.h. ohne dass der Modellierer manuell eingreifen und entscheiden muss. Dazu ist die Definition von Parametern denkbar, die unterschiedliche Vorgehensweisen für bestimmte Konfliktsituationen ermöglichen und die Kombination der Änderungsoperationen entsprechend steuern.

Um die Konsistenz eines Basisprozesses und der definierten Optionen zu prüfen, muss nicht nur die Anwendung einzelner Optionen auf den Basisprozess wieder ein konsistentes Prozessmodell ergeben. Für alle gültigen Kontextkombinationen ist die gegebenenfalls kombinierte Anwendung bzw. Anwendbarkeit mehrerer Optionen zu prüfen. Da die manuelle Überprüfung dadurch sehr aufwendig sein kann, ist eine automatische Konsistenzprüfung notwendig. Für eine effiziente Prüfung werden allerdings Verfahren benötigt, die den Prüfraum geeignet einschränken, da die Anzahl der Kontextkombinationen exponentiell steigt.

4 Ausführung von Prozessvarianten

Optionen sollen nicht nur konsistent und korrekt modelliert werden, sondern auch in einem Workflow-Management-System (WfMS) ausführbar sein. Dies führt zu weiteren Anforderungen, die im Folgenden vorgestellt werden.

4.1 Instanziierung

Auf Grund unterschiedlicher Zielsetzungen von Geschäftsprozess- und Workflow-Modellen [Gad05], muss ein Geschäftsprozess in Verbindung mit seiner Implementierung in ein Workflow-Modell transformiert werden [LR99,Sch98]. Je nach Entwicklungsumgebung kann der Geschäftsprozess in einem Modellierungswerkzeug oder nach Export in eine Implementierungsumgebung (z.B. ARIS SOA Architect [IDS07], WebSphere MQ Workflow [IBM05], etc.) entsprechend angepasst werden. Diese Anpassungen beziehen sich auf technische Detaillierungen, wie z.B. Programmaufrufe und Benutzerkommunikation über Masken. Im vorliegenden Kontext müssen bei der Anpassung die Informationen über modellierte Optionen im WfMS erhalten bleiben.

Bei der Umsetzung des Konzepts der Prozessvarianten wird die Abbildung auf kommerzielle Softwaresysteme angestrebt. Hierfür ist zu klären, welche der beschriebenen Anforderungen existierende Prozessmodellierungs-Werkzeuge und WfMS bereits erfüllen, bzw. wie sie hierfür zu erweitern sind.

4.2 Kontextänderung

Je länger die Laufzeit eines Prozesses ist, desto wahrscheinlicher müssen Änderungen vorgenommen werden, die noch während der Ausführung zu berücksichtigen sind. Ein Auslöser stellt z.B. eine Kontextänderung zur Laufzeit dar. In Abschnitt 3.3 wurden die Anforderungen an die Verwendung von Kontexten vorgestellt und die Kontextvariablen *Geschäftsbereich*, *Baureihe* und *Entwicklungsphase* bereits exemplarisch eingeführt. Dabei können die Kontextvariablen *Geschäftsbereich* und *Baureihe* als statisch betrachtet werden, da sie üblicherweise zu Beginn eines Fahrzeugprojektes festgelegt und im Verlauf des Projektes nicht mehr geändert werden. Die Kontextvariable *Entwicklungsphase* hingegen verhält sich dynamisch, da sich ihr Wert im Verlauf des Projekts ändern kann (z.B. Wechsel der Phase von „Entwicklung“ zu „Serienfertigung“). Entsprechend dieser Eigenschaften der Kontextvariablen muss das Auswerten von Kontextbedingungen sowohl statisch als auch dynamisch zur Laufzeit erfolgen können.

Abbildung 11 zeigt ein Beispiel für das statische Einbinden von Optionen. Die Angabe der statischen Werte für die Kontextvariablen *Geschäftsbereich* und *Baureihe* sowie die Angabe der aktuellen *Entwicklungsphase* führt zur Einbindung von Option 1 in das zu instanziiierende Prozessmodell. Option 2 ist nicht gültig und daher grau hinterlegt dargestellt.

Kontextvariablen, die sich dynamisch ändern, können bei der Erzeugung einer Prozessinstanz nicht endgültig festgelegt werden. Das heißt, dass die tatsächlich relevanten Optionen für ein Prozessmodell erst während der Ausführung

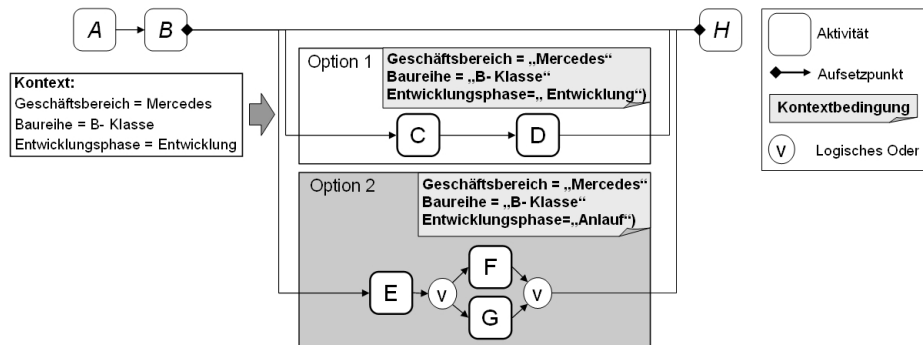


Abb. 11. Statisches Einbinden von Option 1 (da im aktuellen Kontext gültig)

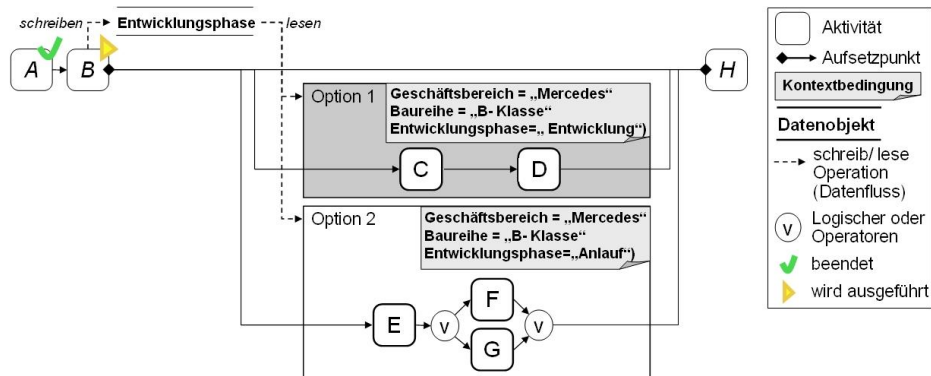


Abb. 12. Dynamisches Einbinden von Option 2 während der Ausführung

eingebunden werden können. Dieses dynamische Einbinden ist in Abbildung 12 beispielhaft dargestellt. Die Kontextvariable *Entwicklungsphase* wird als Datenobjekt behandelt. Während der Ausführung von Aktivität B wird dieses Objekt mit dem Wert „Anlauf“ geschrieben. Am Aufsetzpunkt der Optionen 1 und 2 wird der Wert der Kontextvariablen gelesen und die Kontextbedingungen der Optionen ausgewertet. Dies führt zur Entscheidung für Option 2. Option 1 ist in diesem Kontext nicht gültig und wird daher grau hinterlegt dargestellt. Kontextänderungen, die während der Ausführung einer Option auftreten, sind schwierig zu handhaben. Es muss definiert werden, wie auf sie angemessen reagiert werden kann. Wird ein Wechsel der Optionen während der Ausführung generell erlaubt, muss von der Ausführungs-Steuerung geprüft werden, ob ein Umschalten auf eine andere Option möglich ist, ohne die Konsistenz der Prozessinstanz zu beeinträchtigen. Gegebenenfalls müssen hierfür bereits ausgeführte Aktivitäten kompensiert werden oder die Kontextänderung wird erst für spätere Optionen wirksam.

5 Verwandte Arbeiten

Trotz ihrer hohen Praxisrelevanz finden Prozessvarianten in der Literatur bislang kaum Beachtung. Insbesondere werden die im vorliegenden Beitrag adressierten Anforderungen nicht aufgegriffen. So werden in [LS06] ausmodellierte Prozessvarianten in einem Varianten-Repository verwaltet. Mit Hilfe von Such-Algorithmen können diejenigen Prozessvarianten gefunden werden, die der Struktur der Suchanfrage (in Form eines Prozessfragments) ähneln. Im Gegensatz dazu sollen bei unserem Ansatz zum Einen die Prozessvarianten nicht ausmodelliert und zum Anderen auch nicht getrennt vom Basisprozess verwaltet werden.

Bei PESOA [BBG⁺05,PSWW05] finden sich Beispiele für die Darstellung von Prozessvarianten in UML-Modellen. Es werden Variabilitätsmechanismen, wie z.B. Vererbung, Parametrisierung und Templates erläutert und in verschiedenen Modelltypen beispielhaft angewendet. Dabei liegt der Fokus auf der logischen Konstruktion von Varianten mit UML-Elementen und weniger auf Usability-Aspekten. Außerdem wird die von uns geforderte Verwendung von Kontexten und die Ausführung der Prozessmodelle in Workflow-Management-Systemen nicht berücksichtigt.

Das Konzept der konfigurierbaren Ereignis-Prozess-Ketten (C-EPCs) [RvdA07] ermöglicht das Erstellen von Varianten auf Grundlage eines konfigurierbaren Referenzprozesses. Hier können Funktionen so konfiguriert werden, dass sie entweder Teil der Prozessvariante sein müssen, ausgelassen werden müssen oder bedingt auszulassen sind. Dazu werden die Funktionen mit Informationen hinterlegt, die dem Modellierer bei der Entscheidung helfen. Dies sind zum Einen Richtlinien (ähnlich der Kontextbedingungen) und zum Anderen Anforderungen (Beziehungen zwischen konfigurierten Funktionen). Der Ansatz beschränkt sich zur Zeit noch auf den Kontrollfluss, daher wird die Anpassung von Attributen, wie in unserem Ansatz, nicht berücksichtigt. Auch ein Verschieben von Funktionen ist in diesem Ansatz nicht möglich.

Die im Beitrag eingeführten Aufsetzpunkte sind in ähnlicher Form in der Literatur zu Softwareproduktfamilien beschrieben [SvGB05,DSNB04,BBG⁺05]. Hier werden sie jedoch als „Variation Points“ bezeichnet, über deren Typ (angegeben durch Kardinalitäten) bestimmt wird, wie die im „Variation Point“ modellierten Varianten verwendet werden können. In unserem Ansatz wird die Verwendung von Optionen über die Auswertung von Kontextbedingungen bestimmt. Dadurch sind Optionen immer „optional“, bis eine konkrete Angabe des aktuellen Kontextes für ein zu instanziiertes Prozessmodell erfolgt.

Eine Möglichkeit zur Verwendung von Kontexten zur Individualisierung von Prozessmodellen ist in [Rup02] beschrieben. Mit Hilfe von Projektrahmenbedingungen (Kontexten) und Regeln werden hier halbautomatisch Prozessmodell-Vorschläge generiert. Der Ansatz betrachtet jedoch ausschließlich die Prozessmodellierung. Prozesse mit dynamischen Kontexten, die erst während der Ausführung festgelegt werden können (wie in unserem Beitrag gefordert), bleiben unberücksichtigt.

Einen Ansatz zur flexiblen Ausführung von Workflow-Modellen bietet das Worklets-Konzept [AtHEvdA06]. Hier wird durch dynamisch gebundene Sub-Prozesse

(sogenannte Worklets) eine sukzessive Instanziierung von hierarchischen Workflow-Modellen ermöglicht. Auf diese Weise könnte das im vorliegenden Beitrag geforderte dynamische Einbinden von Optionen umgesetzt werden.

Alle hier beschriebenen Ansätze beziehen sich nur auf einzelne der vorgestellten Anforderungen. Varianten von Geschäftsprozessen und Workflows, betrachtet über den gesamten Prozesslebenszyklus, sind nicht Fokus bisheriger Forschungstätigkeiten.

6 Zusammenfassung und Ausblick

Im Prozessmanagement müssen häufig die verschiedenen Prozessvarianten eines Prozesstyps berücksichtigt werden. Dabei ist eine integrierte Modellierung und Darstellung der Prozessvarianten in einem Prozessmodell, z.B. durch Optionen, unbedingt notwendig. Dies stellt hohe Anforderungen an die Realisierung von Prozessvarianten in allen Phasen des Prozesslebenszyklus. Im vorliegenden Beitrag wurden die Anforderungen und erste Konzepte zur Modellierung und Darstellung von Optionen vorgestellt. Des Weiteren wurde die Bedeutung von Kontextvariablen und -bedingungen zum statischen und dynamischen Einbinden von Optionen erläutert. Außerdem wurde die Konsistenz der Prozessmodelle gefordert, um eine fehlerfreie Ausführung in einem Workflow-Management-System zu gewährleisten.

In weiteren Forschungsaktivitäten werden die beschriebenen Ansätze detailliert. Außerdem sollen für weitere Aspekte Ansätze entwickelt werden, so dass ein durchgängiges Konzept für den gesamten Prozesslebenszyklus, von der Modellierung bis zur Ausführung, von Prozessvarianten entsteht.

Literatur

- [AtHEvdA06] M. ADAMS, A. TER HOFSTEDE, D. EDMOND und W. VAN DER AALST: *Worklets: A Service-Oriented Implementation of Dynamic Flexibility in Workflows*. In: R. MEERSMAN und Z. TARI (Herausgeber): *On the Move to Meaningful Internet Systems, OTM Confederated International Conferences, 14th Intl. Conf. on Cooperative Information Systems*, Band 4275, Seiten 291–308, Berlin, 2006. Springer-Verlag.
- [BBG⁺05] J. BAYER, W. BUHL, C. GIESE, T. LEHNER, A. OCAMPO, F. PUHLMANN, E. RICHTER, A. SCHNIEDERS, J. WEILAND und M. WESKE: *PESOA - Process Family Engineering - Modeling variant-rich processes*. Technischer Bericht 18/2005, Hasso-Plattner-Institut, Potsdam, 2005.
- [BOC07] BOC GMBH: *Das Geschäftsprozessmanagement-Werkzeug ADONIS*, 2007.
- [DSNB04] S. DEELSTRA, M. SINNEMA, J. NIJHUIS und J. BOSCH: *COSVAM: A Technique for Assessing Software Variability in Software Product Families*. In: *Proc. of Int. Conf. on Software Maintenance*, 2004.
- [DvdAtH05] M. DUMAS, W. VAN DER AALST und A. TER HOFSTEDE (Herausgeber): *Process-aware information systems*. Wiley, Department of Information Studies, Uni of California, Los Angeles Los Angeles, CA 90095-1520, 2005.

- [Gad05] A. GADATSCH: *Grundkurs Geschäftsprozess-Management*. Vieweg Verlag, 2005.
- [GHS95] D. GEORGAKOPOULOS, M. HORNICK und A. SHETH: *An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure*. Distributed and Parallel Databases, 3:119–153, 1995.
- [HBR07] A. HALLERBACH, T. BAUER und M. REICHERT: *Managing Process Variants in the Process Life Cycle*. Technischer Bericht TR-CTIT-07-87, Uni Twente, NL, 2007. ISSN 1381-3625.
- [HBR08] A. HALLERBACH, T. BAUER und M. REICHERT: *Modellierung und Darstellung von Prozessvarianten in Provop*. In: *Proc. of Modellierung 2008*, 2008.
- [IBM05] IBM: *WebSphere MQ Workflow - Concepts and Architecture*, 8 Auflage, 2005.
- [IBM07] IBM CORPORATION: *IBM WebSphere Business Modeller, Version 6.0.2*, 2007.
- [IDS06] IDS SCHEER AG: *ARIS Platform Method 7.0*, Oktober 2006.
- [IDS07] IDS SCHEER AG: *ARIS SOA Architect*, 2007. www.ids-scheer.com.
- [KR96] M. KAMATH und K. RAMAMRITHAM: *Correctness issues in workflow management*. In: *Distrib. Syst. Engng 3*, Seiten 213–221. Uni Massachusetts, 1996.
- [LR99] F. LEYMANN und D. ROLLER: *Production Workflow: Concepts and Techniques*. Prentice Hall PTR, September 1999.
- [LS06] R. LU und S. SADIQ: *On Managing Process Variants as an Information Resource*. Technischer Bericht No. 464, School of Information Technology & Electrical Engineering and University of Queensland, Brisbane, Juni 2006.
- [MELS95] K. MISUE, P. EADEST, W. LAI und K. SUGIYAMA: *Layout Adjustment and the Mental Map*. Visual Languages and Computing, 6:183–210, 1995.
- [PSWW05] F. PUHLMANN, A. SCHNIEDERS, J. WEILAND und M. WESKE: *PESOA - Variability Mechanisms for Process Models*. Technischer Bericht 17/2005, Hasso-Plattner-Institut, Potsdam, 2005.
- [Rup02] C. RUPPRECHT: *Ein Konzept zur projektspezifischen Individualisierung von Prozessmodellen*. Doktorarbeit, Uni Fridericiana Karlsruhe, 2002.
- [RvdA07] M. ROSEMAN und W. VAN DER AALST: *A configurable reference modelling language*. Information Systems, 32:1–23, 2007.
- [Sch98] A.-W. SCHEER: *ARIS- Modellierungsmethoden, Metamodelle, Anwendungen*. Springer-Verlag, 1998. Vierte Auflage.
- [SvGB05] M. SVAHNBERG, J. VAN GURP und J. BOSCH: *A taxonomy of variability realization techniques*. Software - Practice and Experience, 35:705–754, 2005.
- [vdA97] W. VAN DER AALST: *Verification of workflow nets*. In: W. VAN DER AALST (Herausgeber): *Application and Theory of Petry Nets*, Band LNCS 1248, Seiten 407–426. Springer-Verlag Berlin Heidelberg, Juni 1997.
- [vdAHW03] W. VAN DER AALST, A. HOFSTEDE und M. WESKE: *Business process management: A survey*. In: W. VAN DER AALST, A. HOFSTEDE und M. WESKE (Herausgeber): *Proc. of BPM 2003*, Band LNCS 2678, Seiten 1–12. Springer-Verlag Berlin Heidelberg, 2003.
- [Wes07] M. WESKE: *Business Process Management - Concepts, Languages, Architectures*. Springer, 2007.

- [WRR07] B. WEBER, S. B. RINDERLE und M. U. REICHERT: *Change Patterns and Change Support Features in Process-Aware Information Systems*. In: *Proceedings 19th International Conference on Advanced Information Systems Engineering (CAiSE 2007)*, Trondheim, Norway, Band 4495 der Reihe *Lecture Notes in Computer Science (LNCS)*, Seiten 574–588, London, June 2007. Springer Verlag.
- [WWRD07] B. WEBER, W. WILD, M. U. REICHERT und P. DADAM: *ProCycle - Integrierte Unterstützung des Prozesslebenszyklus*. Technischer Bericht 4 / 2007, Bremen, November 2007.