

# Modellierung von Flexibilität mit Ereignisgesteuerten Prozessketten (EPK)

Jürgen Grief\*, Heinrich Seidlmeier\*\*

\*BC & S GmbH  
Bank Consulting & Solutions  
Schumanstr. 33  
52146 Würselen  
J.Grief@bcs-gmbh.de

\*\*Fachhochschule Rosenheim  
Institut für Organisation und Wirtschaftsinformatik  
am Fachbereich Betriebswirtschaft  
Hochschulstr. 1  
83024 Rosenheim  
seidlmeier@fh-rosenheim.de

**Abstract:** „Flexibilität“ oder auch „Adaptivität“ ist, gleichermaßen in Theorie und Praxis, in Zeiten der beschleunigten Ökonomie einer der betrieblichen Erfolgsfaktoren schlechthin. Diese Eigenschaft eines Unternehmens, das dann als „agil“ gilt, wird auch beim Management von Geschäftsprozessen gefordert. Auf der strategischen Ebene finden sich einschlägige Hinweise in der wissenschaftlichen Literatur, in Marketingprospekten oder in Firmenbroschüren in großer Zahl. Auf der „tiefen“ Ebene der detaillierten Prozessmodellierung zeigt sich ein anderes Bild. Ausführungen zur konkreten Modellierung von flexiblen Prozessen sind eher selten. Hier setzt dieser Beitrag an. Es werden Konstruktionsideen für flexible Prozesse entwickelt und mit EPK umgesetzt. Die grundlegenden Ideen „Modularisierung“ und „Entkopplung“ führen zur selbststeuernden Prozesskonfiguration und variablen Funktionsreihenfolge. Damit wird Flexibilität zwischen und in entkoppelten Prozessmodulen erzeugt.

## 1 Bedeutung und Definition von Flexibilität

Schon seit Jahren unstrittig ist sicherlich die Bedeutung der Flexibilität für den Unternehmenserfolg [z.B. GSW98], insbesondere in kleineren und mittelständischen Unternehmen [z.B. Fe86]. Unter anderem wird die Fähigkeit flexibel (oder auch adaptiv bzw. agil) zu sein, als das wichtigste Ziel des Geschäftsprozessmanagements erachtet [Jo05, S. 9]. Weiterhin ergab eine Umfrage unter mehr als 4000 Führungskräften weltweit, dass die Fähigkeit, Flexibilität tatsächlich auch in Unternehmenswandel umzusetzen, als größte Managementtherausforderung bis 2010 betrachtet wird [Ec2005].

Insbesondere ist beispielsweise im Bankensektor durch die rapide zunehmenden Marktanteile der Transaktionsbanken der Bedarf an flexiblen Referenzprozessen deutlich gewachsen. Die jeweilige Transaktionsbank definiert einen Referenzprozess, welcher dann mandantenspezifisch anzupassen ist. Diese flexiblen Referenzprozesse stellen einen wesentlichen Faktor für die Marktposition der Transaktionsbanken dar. Der Erfolg der Transaktionsbanken legt den Schluss nahe, dass in naher Zukunft weitere „Dienstleistungsfabriken“ in anderen Geschäftsfeldern entstehen werden, und somit die Bedeutung flexibler (Referenz-)Prozesse zunehmend steigen wird.

Trotz der hier nur angerissenen Wichtigkeit wird die tiefere Auseinandersetzung mit dem vielgenannten Phänomen „Flexibilität“ oft vernachlässigt, gerade auch auf Prozessmodellierungsebene. Die eigentlich zuständige klassische (deutschsprachige) Organisationsliteratur setzt sich nur sehr oberflächlich damit auseinander [z.B. KK92 und Sc03], ebenso die älteren und gleichermaßen neueren Prozessmanagement-Standardwerke [z.B. HC94 und BKR05].

Flexibilität wird sehr unterschiedlich definiert. Generell kann man unter Flexibilität (im Sinne von Adaptivität) die Fähigkeit verstehen, sich ändernden Rahmenbedingungen anpassen zu können [A198, S. 2]. Mit Betonung des Zeitfaktors ist ein System dann flexibel, „wenn einem Wandlungsbedarf ein in angemessener Zeit aktivierbares Wandlungspotential im System gegenübersteht“ [AS04, S. 70]; Flexibilität ist folglich als Wandlungsfähigkeit zu verstehen. Neben dem Zeitfaktor ist ferner der Anpassungsaufwand zu berücksichtigen. Ein Prozess ist nur dann flexibel, wenn ein geringer Wandlungsbedarf nicht die Aktivierung eines diesem Bedarf unangemessenen Wandlungspotenzials erfordert.

Diese Anpassungs- bzw. Wandlungsfähigkeit wird auch diesem Beitrag zugrunde gelegt. Vor diesem Hintergrund werden im folgenden Kapitel verschiedene allgemeingültige Ansätze zur flexiblen Gestaltung von Prozessen kurz dargestellt. Es wird sich zeigen, dass alle Vorschläge im Kern auf wenigen gemeinsamen Grundideen aufbauen. Diese Kerngedanken werden im darauf folgenden Kapitel aufgegriffen, weiterentwickelt und als Basis für flexible Ereignisgesteuerte Prozessketten verwendet.

## **2 Allgemeine Ansätze zur Bewältigung von Umweltänderungen in Prozessen**

Es existiert eine ganze Reihe von Möglichkeiten, um den Anforderungen gerecht zu werden, die aus Umweltänderungen resultieren. Nachfolgend werden einige davon kurz dargestellt.

## **2.1 Erzeugung von Prozessvarianten**

Übersichtsartig berichtet Allweyer über „Prozesshandbücher“, „Prozessbibliotheken“ und „Prozesspartikel“ [Al98, S. 67 f. und die dort ausgewertete Literatur]. In einem Prozesshandbuch werden Teilprozesse und Regeln für deren Verknüpfung abgelegt. Auf dieser Basis lassen sich, wenn Wandlungsbedarf entsteht, Prozessalternativen erzeugen. Allerdings gibt es keine Angaben, welche Alternativen zielgerecht unter den gegebenen Rahmenbedingungen zu wählen sind. Eine Prozessbibliothek enthält wieder verwendbare Referenzprozessbausteine. Regeln bzw. Kriterien zur Ermittlung der geeigneten Bausteine werden im Einzelnen nicht beschrieben. Im Rahmen der Verwendung von Prozesspartikeln zur Prozessgestaltung werden die Anwendungsvoraussetzungen für die Partikel ausdrücklich behandelt. Zentral ist ein feststehendes, so genanntes „Essentielle Modell“ mit den zur Zielerreichung unbedingt notwendigen Prozessschritten und Abhängigkeiten. Aufgrund von Umweltveränderungen notwendige Prozessvarianten werden auf Basis des essentiellen Modells und passender generischer Prozesspartikel modelliert. Da die Einsatzvoraussetzungen von Prozesspartikeln definiert sind, können geeignete Prozessalternativen ausgewählt werden.

Da in allen genannten Fällen (Handbuch, Bibliothek, Partikel) letztlich Prozessvarianten erzeugt werden, spielt das „Management“ dieser Varianten, bevorzugt durch entsprechende Modellierungswerkzeuge, eine wichtige Rolle.

Die effiziente Handhabung von Prozessvarianten stellt eine wesentliche Grundlage für den Erfolg von Referenzprozessen dar. Ausgehend von einem branchenspezifischen Referenzprozess wird ein unternehmensspezifischer Prozess generiert, welcher allgemein als Prozessvariante zu interpretieren ist. Führt eine Dienstleistungsfabrik (Beispiel: Transaktionsbank) die Prozesse mehrerer Unternehmen (derselben Branche) durch, so muss der Referenzprozess derart flexibel gestaltet sein, dass aus ihm mit vertretbarem Aufwand unternehmensspezifische Prozesse abgeleitet und diese mit ihm synchron gehalten werden können.

## **2.2 Verwendung „robuster“ Modellierungsmethoden**

Weitere Vorschläge zur prozessorientierten Bewältigung von Umweltänderungen sind die objektorientierte und die ressourcenbasierte Modellierung.

Die Wurzeln der objektorientierten Softwareentwicklung reichen bis in die 1970er Jahre zurück. Ab den 90er Jahren setzt sich dieses Paradigma zunehmend gegen die prozedurale Vorgehensweise durch [Oe2003, S. 12]. Die objektorientierte Modellierung von Prozessen, insbesondere mit der Unified Modeling Language (UML), ist in etwa ab Mitte der 90er bekannt [Zi99, S. 65 ff.]. Weiterhin existieren Vorschläge zu „Objektorientierten Ereignisgesteuerten Prozessketten“ (oEPK) [SNZ97], auch in Kombination mit der UML [LA98; Da99].

Gerade durch die Zusammenführung von objektorientierter, eher umsetzungstechnischer („UML“) und prozessorientierter, eher betriebswirtschaftlich-konzeptioneller Sichtweisen („EPK“) verspricht man sich, die Lücken und Schwächen des jeweils einen Ansatzes durch die Integration mit dem jeweils anderen Ansatz zu umgehen. Eine Gegenüberstellung der wesentlichen Elemente von UML Aktivitätsdiagrammen und EPK findet sich in [Gr05].

Die an Unternehmensressourcen ausgerichtete Prozessmodellierung ist eine sehr neue Sichtweise [BMH05; Se05]. Der „Resource-Based View of the Firm“ führt den Erfolg eines Unternehmens auf Ressourcenasymmetrien zurück. Dauerhaft überdurchschnittliche Gewinne und nachhaltige, verteidigungsfähige strategische Wettbewerbsvorteile als deren Ursache erklären sich demnach aus einer überlegenen Ressourcenausstattung und/oder –nutzung [grundlegend Ba91].

Prozesse werden dabei als Aneinanderreihung von ressourcenbasierten Prozessbausteinen betrachtet. Ein Prozessbaustein besteht aus einer Funktion (= Prozessschritt), dazu notwendigen Ressourcen (beispielsweise Sachmittel) und Fähigkeiten zur Bewältigung der Funktionsdurchführung. Auf der Grundlage dieser vorhandenen Bausteine können dann Prozesse nach (Wandlungs-) Bedarf zusammengestellt werden.

Da in beiden Fällen mit „Objekten“ und „Ressourcen“ eher robuste Prozessbestandteile<sup>1</sup> im Vordergrund stehen, sind darauf aufbauende Modelle weniger sensibel gegenüber Umweltänderungen und können ohne Modifikationen weiterverwendet werden. Derartige Prozesse sind damit nicht im eigentlichen Sinne flexibel, können aber bevorzugt in dynamischen Umwelten eingesetzt werden.

### **2.3 Wahl des geeigneten Abstraktionsgrades**

Unter Abstraktionsgrad soll die Detaillierung, auch Granularität der Prozessmodellierung verstanden werden. Es handelt sich nicht um einen eigenständigen Ansatz, sondern um ein grundlegendes Konzept. Allgemein gilt: Je feiner ein Prozess modelliert ist, desto genauer ist sein Ablauf festgelegt. Damit fehlen Freiheitsgrade in der Abarbeitung und reduzieren als Folge die in Kapitel 1 angesprochenen Wandlungspotentiale. Für sehr abstrakt formulierte Prozesse gilt das Gegenteil.

Besteht beispielsweise der Prozess „Rechnung bearbeiten“ nur aus einem, gleichnamigen Prozessschritt, kann die Bearbeitung einer Rechnung sehr flexibel erfolgen – im Bedarfsfall z.B. weniger genau, bevorzugt oder beschleunigt. Zur eigentlichen Aufgabe „Rechnung bearbeiten“ existiert keine weitergehende Bearbeitungsanweisung. Eine sehr genaue Ablaufbeschreibung ver- oder zumindest behindert flexible Vorgehensweisen. Von zentraler Bedeutung ist die Bestimmung des „optimalen“ Abstraktionsgrades.

---

<sup>1</sup> Vgl. zur Prozesseigenschaft „Robustheit“ [A198, S. 89 ff.].

Damit im Falle von sehr groben Prozessbeschreibungen vorgegebene Prozessziele trotzdem erreicht werden (Flexibilität damit nicht opportunistisch ausgenutzt wird), bieten sich verschiedene organisatorische Lösungen an [PDF05, S. 244 f.]. Z.B. kann bei der Auswahl der entsprechenden Mitarbeiter auf das Vorhandensein notwendiger Werte (Genauigkeit, Loyalität, Zuverlässigkeit u.ä.) geachtet werden.

Die Entscheidungsmöglichkeiten, die eine grob-granulare Prozessbeschreibung hinsichtlich der manuellen Durchführung eines Prozesses bietet, setzen Kenntnisse bezüglich möglicher Aktionen und Effekte voraus. Dieses „Prozesswissen“ des Ausführenden muss auch in die Beschreibung automatischer Prozesse integriert werden, damit ausführende Systeme in analoger Weise reagieren können.

## **2.4 Zusammenfassende Erkenntnisse**

Zunächst fällt auf, dass sehr viele der in den vorigen Abschnitten genannten Ansätze mit Teilprozessen, Prozessbausteinen u.ä. sowie sich daraus ergebenden Varianten arbeiten. Allerdings ist die Bildung von Teilprozessen, und damit verbunden auch die Wahl des optimalen Abstraktionsgrades, nicht durchgängig gelöst. Auch die richtige Verknüpfung der Teilprozesse ist nicht in allen Fällen klar. Diese methodische Lücke ist bei der zielgerechten Wahl von Alternativen, unter den gegebenen Rahmenbedingungen, sogar noch größer. Interessant erscheint, weil in den beschriebenen Ansätzen vernachlässigt, die methodische Überlegung, aus Teilprozessen automatisch Gesamtprozesse zu erzeugen.

## **3 Flexible Prozesse durch Modularisierung und Entkopplung**

Der in diesem Absatz dargestellten Konstruktion flexibler Prozesse, formuliert als (weitergehend zu diskutierende) „Konstruktionsideen“, liegt folgende Basisüberlegung zu Grunde. Flexibilität in Organisationen wird generell durch Modularisierung von organisatorischen Strukturen und Prozessen sowie durch die Entkopplung dieser Module erreicht. Entkoppelte Module erleichtern Neukonfigurationen von Strukturen und Prozessen und erhöhen somit die organisatorische Flexibilität [ähnlich AS04, S. 70 f.].<sup>2</sup> Die konkrete Umsetzung dieser Ideen mit EPK erfolgt im folgenden Kapitel 4.

---

<sup>2</sup> Diese auf organisatorische Aspekte ausgerichtete Sichtweise kommt grundsätzlich auch zur Anwendung bei der Konstruktion flexibler Softwarearchitekturen [SK03].

### **Konstruktionsidee 1: Modularisierung**

Als erste grundlegende Idee wird vorgeschlagen, Prozesse aus Modulen aufzubauen. Ein Modul besteht aus Schnittstellen und einem Kern. Der Kern setzt sich aus grundsätzlich freien Bausteinen mit der eigentlichen Funktionalität<sup>3</sup> zusammen. Diese Module können auch als Ressourcen [Se05] oder Services [BMH05] verstanden werden.

### **Konstruktionsidee 2: Entkopplung der Module**

Die Flexibilisierung von Prozessen und damit die Erzeugung von Wandlungspotential werden durch die Entkopplung der Module erreicht. Es entstehen, als zweite wesentliche Idee, autonome Prozessmodule (bzw. Teilprozesse), die situativ, aber zielgerecht zusammengesetzt werden können.

### **Konstruktionsidee 3: Modul-Intra- und –Interflexibilität**

Modularisierung und Entkopplung erzeugen Flexibilität auf zwei Ebenen:

- Innerhalb der Module („Modul-Intraflexibilität“): Die freien (bzw. auch entkoppelten) Prozessbausteine können in einem Modul grundsätzlich beliebig kombiniert werden.
- Zwischen den Modulen („Modul-Interflexibilität“): Die entkoppelten Prozessmodule bzw. Teilprozesse können grundsätzlich beliebig zu einem Gesamtprozess kombiniert werden.

### **Konstruktionsidee 4: Selbststeuernde Prozesskonfiguration**

Ein „Kopplungssystem“ enthält die Methodik und die notwendigen Umweltinformationen, um die Prozessmodule selbständig zu Gesamtprozessen zusammenbauen zu können. Diese Idee reduziert die durch Flexibilität gewöhnlich erzeugte Komplexität. Es müssen nicht alle in der Diskurswelt denkbaren Modulkombinationen (als Redundanz erzeugende Varianten) vorgehalten werden – mit der Gefahr einer „kombinatorischen Explosion“. Der von nicht beeinflussbaren Umweltveränderungen ausgelöste Handlungsbedarf wird durch die selbständige Erzeugung der einen passenden Prozessvariante gedeckt. Das Wandlungspotential ist nicht explizit modelliert, sondern liegt v.a. im modul-interflexiblen Kopplungssystem.<sup>4</sup>

---

<sup>3</sup> Vgl. zu den Gestaltungszielen bei der Modularisierung und zum Grad der Modularisierbarkeit [AS04 S. 71f. und S. 74]. Gemäß der ARIS-Methodik [Sc01] setzt sich ein Baustein aus den Sichten Funktion, Organisation, Daten und Leistung zusammen.

<sup>4</sup> Und daneben in der Modul-Intraflexibilität.

Zur methodischen Umsetzung dieser Ideen bieten sich die Ereignisse in EPK an. Zum einen bilden Ereignisse Umweltveränderungen ab. Eine derartige Wandlungsbedarf erzeugende Änderung ist nichts anderes als ein Ereignis. Zum anderen werden nachfolgend Ereignisse zur Umsetzung des oben angesprochenen Kopplungssystems in Form von (ARIS-) Ereignisdiagrammen herangezogen.

#### **Konstruktionsidee 5: „Kernmodell“**

Das Kernmodell enthält zwingend notwendige Prozessmodule und schränkt dadurch die Flexibilität bei der Prozessgestaltung ein. Die Notwendigkeit eines derartigen Kernmodells kann sich aus fixierten Unternehmensvorgaben bzw. Prozesszielen (z.B. „100%-Endkontrolle“) oder aus vorhandenen rechtlichen, betrieblichen oder sonstigen Verordnungen (z.B. „Vier-Augen-Prinzip“) ergeben. Weiterhin erzeugt die Verwendung eines Kernmodells „kontrollierte“ Flexibilität, indem es ein „Ausufern“ (z.B. übermäßiges bzw. unwirtschaftliches Erfüllen von Kundenänderungswünschen) bzw. sogar im Extremfall Willkür verhindert.

## **4 Umsetzung der Konstruktionsideen mit EPK**

Die Konstruktionsidee der selbststeuernden Prozesskonfiguration wird im ersten Teil dieses Kapitels auf der Basis von EPK und Ereignisdiagrammen umgesetzt. Die auf diesem Wege erzielte Flexibilisierung von Prozessdarstellungen führt zu einer signifikanten Verbesserung der Modul-Interflexibilität.

Im Anschluss wird durch ein Konzept zur Abbildung variabler Funktionsreihenfolgen innerhalb von EPK ein Ansatz zur Erhöhung der Modul-Intraflexibilität vorgestellt.

### **4.1 Selbststeuernde Prozesskonfiguration**

In der EPK-Literatur wird der Ereignisbegriff sehr weit gefasst. Chen und Scheer definieren in dem Grundsatzartikel [CS94] Ereignisse wie folgt: „Ereignisse in der EPK lösen Funktionen aus und sind deren Ergebnis. Sie repräsentieren zugleich das Ende einer Funktionsausführung und den Ausführungsbeginn der Nachfolger-Funktion. [...]“. Dieser Ereignisbegriff entspricht in seinem Kern auch der Definition in [KNS92]. Rump verwendet zwar in [Ru99] den Ereignisbegriff aus [KNS92], er weist jedoch darauf hin, dass er ihn letztlich im Sinne einer Zustandsdefinition nutzt. Davis folgt dem Ereignisbegriff aus [KNS92] in [Da03] weitgehend, wobei er zwischen Startereignissen, mittleren Ereignissen und Endereignissen unterscheidet.

Uthmann geht in seiner Ereignisdefinition in [Ut97] einen Schritt weiter: „Ereignisse bezeichnen Zustandsübergänge, wobei Bereitstellungs- und Auslöseereignisse unterschieden werden, die sich jeweils auf die Bereitstellung von Objekten als Ergebnis einer ausgeführten Funktion bzw. auf das Eintreten einer Objektkonstellation beziehen, die zur Auslösung einer Funktion führt. In der Konsequenz beginnt jede Prozeßkette mit einem oder mehreren Auslöseereignissen, die eine prozeßinstanzierende Anfangsbedingung bzw. mehrere -teilbedingungen repräsentieren, und endet mit einem oder mehreren Bereitstellungsereignissen, die das Eintreten des Zustandes nach Prozeßende bezeichnen.“.

Während Uthmann lediglich Bereitstellungs- und Auslöseereignisse unterscheidet<sup>5</sup>, werden wir eine Kategorisierung in drei verschiedene Ereignistypen vornehmen, auf deren Grundlage die selbststeuernde Prozesskonfiguration umgesetzt wird. Auf der Basis des Ereignisbegriffs aus [CS94] seien folgende Ereignistypen definiert:

- Ein Bereitstellungsereignis ist ein Ereignis, das unmittelbar aus einer Funktionsausführung resultiert.
- Ein internes Auslöseereignis ist ein Ereignis, welches im Rahmen des Prozessablaufs die unmittelbare Konsequenz von Bereitstellungsereignissen oder externen Auslöseereignissen (oder Kombinationen dieser Ereignisse) ist, und – ggf. in Verbindung mit weiteren (internen oder externen) Auslöseereignissen – zu der Ausführung einer oder mehrerer Funktionen führt.
- Ein externes Auslöseereignis ist ein Ereignis, welches nicht aus dem Prozessverlauf resultiert, sondern außerhalb des betrachteten Prozesses ausgelöst worden ist, und – ggf. in Kombination mit weiteren (internen oder externen) Auslöseereignissen – zu der Ausführung einer oder mehrerer Funktionen führt.

Die strikte Unterscheidung von Bereitstellungs- und Auslöseereignissen ermöglicht es, den Prozessverlauf von den einzelnen Teilprozessen zu entkoppeln. Während die einzelnen Teilprozesse in EPK abgebildet werden, wird der Prozessverlauf in Ereignisdiagrammen modelliert. Dabei wird folgenden Prinzipien gefolgt:

- In einem Ereignisdiagramm wird durch eine gerichtete Verbindung von Ereignis e1 zu Ereignis e2 ausgedrückt, dass aus dem Bereitstellungsereignis oder dem externen Auslöseereignis e1 das interne Auslöseereignis e2 folgt.
- Bei den Startereignissen einer EPK handelt es sich ausschließlich um (interne oder externe) Auslöseereignisse.
- Ist ein von einer Funktion ausgelöstes Ereignis im Rahmen des Prozessverlaufs unweigerlich auch das auslösende Ereignis der nachfolgenden Funktion, so repräsentiert dieses sowohl ein Bereitstellungs- als auch ein internes Auslöseereignis, und wird nicht in einem Ereignisdiagramm aufgeführt.

---

<sup>5</sup> Diese Unterscheidung wird auch in [BKR05] diskutiert.



Das für die Idee der selbststeuernden Prozesskonfiguration wesentliche Kopplungssystem wird demzufolge vollständig in Ereignisdiagrammen abgebildet. Die daraus resultierende lose Kopplung der EPK führt zu einer weitgehenden Modulautonomie und infolgedessen Modul-Interflexibilität, da die Bedingungen bezüglich der Modulreihenfolge nicht in die EPK integriert sind.

Die Unterscheidung von Auslöse- und Bereitstellungsereignissen führt in Kombination mit den explizit formulierten Prinzipien zur Erstellung von Ereignisdiagrammen nicht nur zu Modul-Interflexibilität, sondern bildet auch die Basis für ein Vorgehensmodell zur Erstellung flexibler Prozessmodelle.

Ein Gesamtprozess besteht aus mehreren Teilprozessen (Modulen). Zwei Teilprozesse p1 und p2 sind durch ein gemeinsames Ereignis e1 verbunden. Das Ereignis e1 wird in der Folge nur noch als Bereitstellungsereignis, nicht mehr jedoch als Auslöseereignis klassifiziert. Es wird ein neues Auslöseereignis e2 eingeführt, welches e1 als Starterereignis des Teilprozesses p2 ablöst (e1 bleibt unverändert Endereignis von p1). Die Teilprozesse p1 und p2 sind nun durch die explizite Formulierung von Auslöse- und Bereitstellungsereignissen entkoppelt. Es wird ein Ereignisdiagramm angelegt, welches die Transformation des Bereitstellungsereignisses e1 in das Auslöseereignis e2 explizit beschreibt. Dabei können weitere (externe) Auslöseereignisse, welche Umwelteinflüsse beschreiben oder Prozess(ablauf)konfigurationen spezifizieren, in die Transformationsregeln einbezogen werden.

Ein wesentliches Merkmal dieses Vorgehensmodells besteht darin, dass zu Beginn der Modellierung eines Prozesses nicht sämtliche „denkbaren“ Prozessverläufe abgebildet werden müssen. Die hier skizzierte Methodik führt durch die Einführung und kontinuierliche Erweiterung eines Kopplungssystems zu autonomen Teilprozessen und damit zu Modul-Interflexibilität.

Der hier vorgestellte Ansatz beschreibt die Module in EPK. Diese Module werden durch das Kopplungssystem, welches in Ereignisdiagrammen modelliert wird, situationsabhängig miteinander verknüpft.

Die in den Ereignisdiagrammen vorgenommene Abbildung von Bereitstellungs- und externen Auslöseereignissen auf interne Auslöseereignisse kann formal im Sinne wissensbasierter Regelsysteme mit Mitteln der mathematischen Logik beschrieben werden. Damit können in diesem Sinne verwendete Ereignisdiagramme als Regelsysteme interpretiert werden, auf die das Prinzip der Vorwärtsverkettung<sup>6</sup> angewendet wird.

Die Komposition des Gesamtprozesses aus den Teilprozessen (EPK) und dem Kopplungssystem (Ereignisdiagramme) kann durch folgenden Algorithmus skizziert werden:

---

<sup>6</sup> Vgl. zur Vorwärtsverkettung [Bi93] und [GFH90].

1. Bestimmung der Menge aller externen Auslöseereignisse.
2. Bestimmung der Menge aller internen Auslöseereignisse mittels Vorwärtsverkettung auf Basis der Ereignisdiagramme und der vorliegenden Bereitstellungs- und Auslöseereignisse.
3. Bestimmung aller EPK, deren Menge der Startereignisse eine Teilmenge<sup>7</sup> der Menge aller Auslöseereignisse ist (unter Berücksichtigung der durch Konnektoren in den EPK beschriebenen Verknüpfungslogik).
4. Durchlauf aller in Schritt 3 ermittelten EPK. Nach jedem Durchlauf einer EPK wird Schritt 2 wiederholt.

Der Algorithmus terminiert, wenn alle EPK durchlaufen sind und keine weitere EPK durch die Menge der vorliegenden Auslöseereignisse instanziiert werden kann.

Die strikte Trennung der EPK und Ereignisdiagramme nach dem Prinzip der selbststeuernden Prozesskonfiguration bietet zukunftsgerichtet einen weiteren Vorteil: sie ermöglicht die anwendungsfallbezogene<sup>8</sup> Komposition eines Prozessmodells. Während bei der klassischen Darstellung aus bestehenden Modellen Teile entfernt werden müssen, ermöglicht es das modul-interflexible Kopplungssystem, den Gesamtprozess aus den EPK der Teilprozesse (wie hier beschrieben) zusammensetzen; nicht benötigte EPK können einfach weggelassen werden, ohne dass sie modifiziert werden müssen. Damit wird die Basis für die Komposition von Prozessmodellen unter Berücksichtigung konkreter Anwendungsfälle geschaffen.<sup>9</sup>

Das Prinzip der selbststeuernden Prozesskonfiguration soll nun an einem einfachen Beispiel erläutert werden:

---

<sup>7</sup> Sind Startereignisse durch XOR-Konnektoren verknüpft, darf von diesen Startereignissen nur genau eines in der Menge der Auslöseereignisse enthalten sein.

<sup>8</sup> Zu Anwendungsfällen vgl. [Co01].

<sup>9</sup> Wenngleich eine EPK aufgrund ihrer Modul-Intraflexibilität eine Menge ähnlicher Anwendungsfälle abdecken kann.

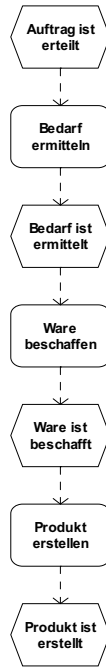


Abbildung 1: Starrer Prozess zur Auftragsbearbeitung

Nach der Modellierung des in Abbildung 1 abgebildeten Prozesses wird festgestellt, dass auftragsabhängig eine Freigabe des Auftrags vor der eigentlichen Auftragsbearbeitung durchzuführen ist. Dies bedeutet, dass das externe Ereignis „Auftrag ist erteilt“ nicht mehr unmittelbar das auslösende Ereignis für den Teilprozess zur Auftragsbearbeitung darstellt. Der gemäß dem Prinzip der selbststeuernden Prozesskonfiguration angepasste Prozess lässt sich durch folgende Modelle abbilden:

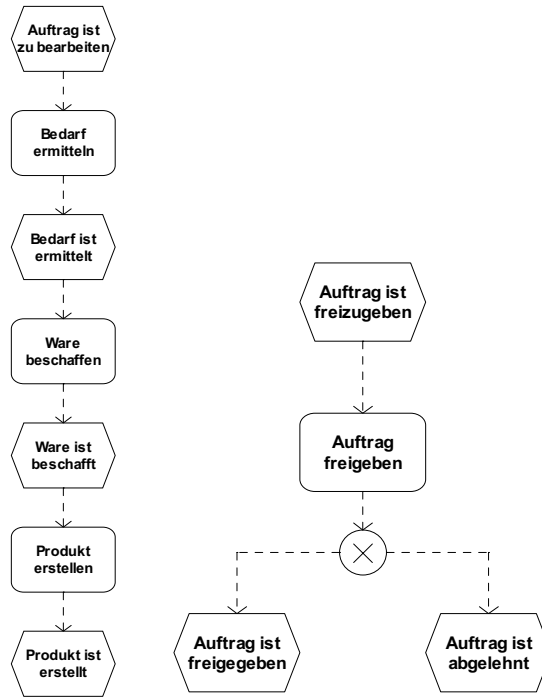


Abbildung 2: Flexibler Prozess zur Auftragsbearbeitung (EPK)

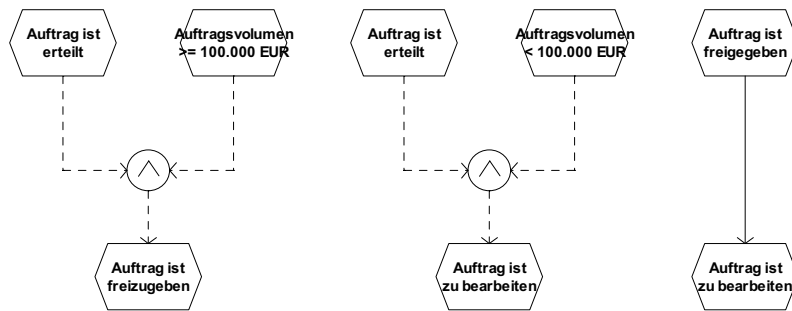


Abbildung 3: Flexibler Prozess zur Auftragsbearbeitung (Ereignisdiagramm)

In der ursprünglichen EPK zur Auftragsbearbeitung wurde lediglich das externe Auslöseereignis „Auftrag ist erteilt“ durch das interne Auslöseereignis „Auftrag ist zu bearbeiten“ substituiert. Ansonsten bleibt diese EPK unverändert. Sie ist völlig unabhängig von der EPK mit der Funktion „Auftrag freigeben“. Jede der beiden EPK enthält somit losgelöst vom Gesamtprozess eine isolierte Darstellung des betreffenden Teilprozesses. Das in dem Ereignisdiagramm abgebildete Kopplungssystem beschreibt die eigentliche Prozesslogik. Hier kann der Gesamtprozess flexibel an sich ändernde Anforderungen bezüglich der Notwendigkeit einer Auftragsfreigabe angepasst werden. Das Ereignisdiagramm beinhaltet damit letztlich die Prozesskonfiguration, welche für die Komposition der autonomen Teilprozesse zum Gesamtprozess verantwortlich ist.

Zum Vergleich seien die beiden klassischen Möglichkeiten in den Abbildungen 4 und 5 aufgezeigt, den Beispielprozess ohne das Konzept der selbststeuernden Prozesskonfiguration darzustellen.

Der entscheidende Nachteil in Abbildung 4 liegt in der engen Kopplung aller möglichen Prozessverläufe sowie der integrierten Abbildung von fachlichen Funktionen und Prozesslogik. Auftragsbearbeitung und -freigabe sind hier nicht als autonome Teilprozesse dargestellt, sondern gemäß der aktuellen Prozesslogik miteinander verknüpft. Sie sind infolgedessen keine autonomen Elemente einer Bibliothek von Teilprozessen.

Ungeachtet dessen, dass der Ablauf einer Freigabe für Aufträge  $\geq 100.000$  EUR in Bezug auf Aufträge mit geringen Beträgen ohne Belang ist, trägt er zur Komplexität der Darstellung des Ablaufs für Aufträge  $< 100.000$  EUR bei. Die integrierte Darstellung der Prozessverläufe hat zur Folge, dass jede Änderung der Prozesslogik (beispielsweise zusätzlich die Berücksichtigung „priorisierter“ Aufträge) eine Anpassung der vorliegenden EPK bedingt, obwohl die Funktionalität der eigentlichen Auftragsbearbeitung davon unberührt bleibt.

Die hier verdeutlichten Probleme werden durch die Einführung von Prozessschnittstellen und die Verteilung der Abläufe auf mehrere EPK nicht grundlegend behoben, wie Abbildung 5 zeigt:

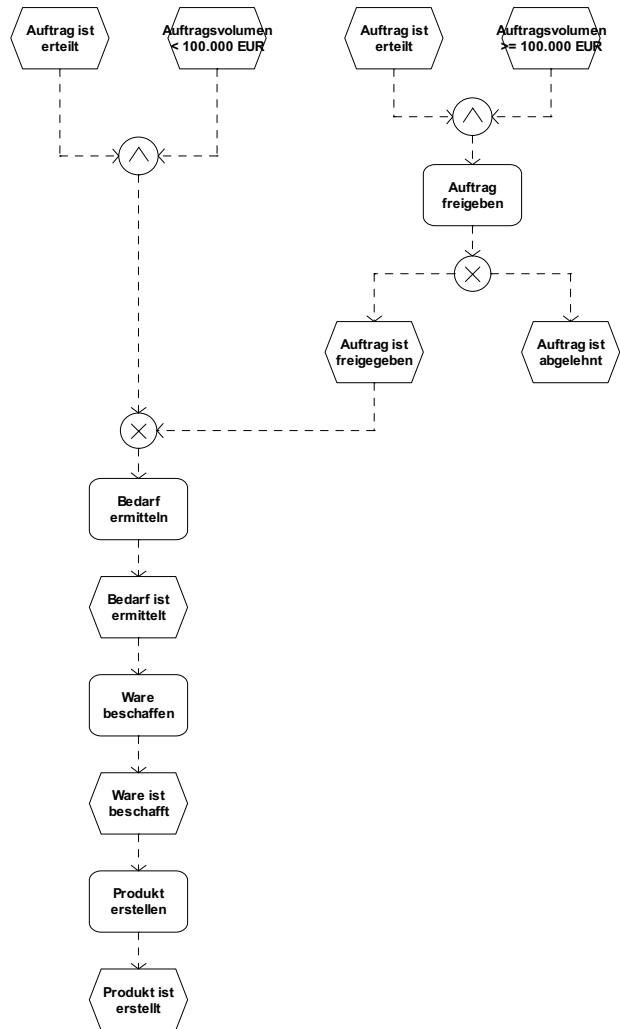


Abbildung 4: Unflexibler Prozess zur Auftragsbearbeitung (eine EPK)

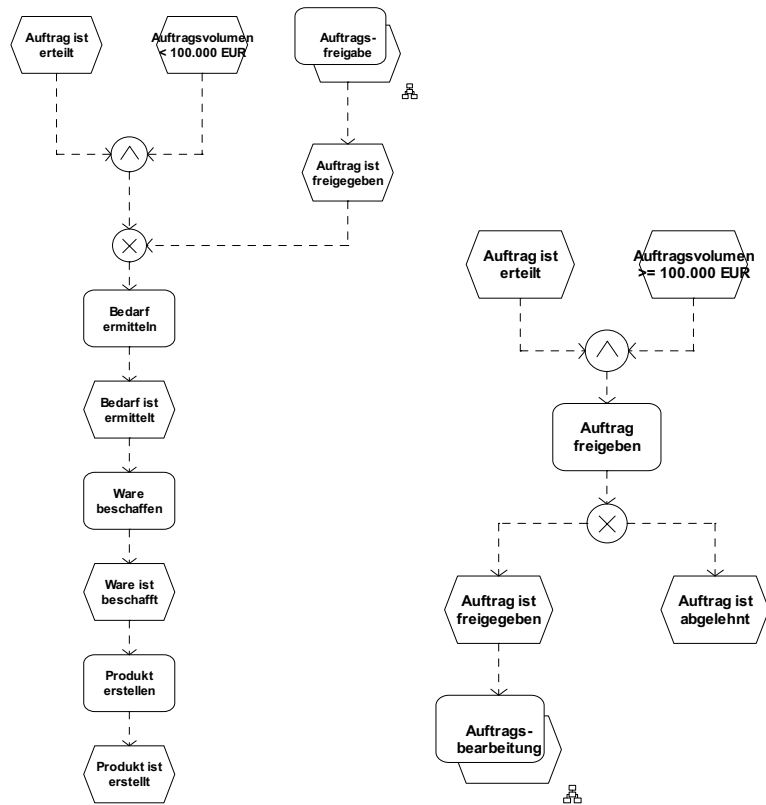


Abbildung 5: Unflexibler Prozess zur Auftragsbearbeitung (zwei EPK)

Durch die Einführung der Prozessschnittstellen wird zwar die interne Logik der Freigabe von der eigentlichen Auftragsbearbeitung getrennt, die EPK zur Auftragsbearbeitung bleibt jedoch weiterhin fest mit der (EPK zur) Freigabe verknüpft. Allgemein erfordert in der klassischen Darstellung eine Änderung der Modulreihenfolge die Anpassung der beteiligten EPK. Die Autonomie der Teilprozesse (EPK) ist durch die Prozessschnittstellen und die gleichzeitige Verwendung der Startereignisse als Endereignisse (und umgekehrt) verletzt.

Die selbststeuernde Prozesskonfiguration sorgt durch die Entkopplung der einzelnen EPK und die Kapselung des Prozessverlaufs in Ereignisdiagrammen für eine erhöhte Modul-Interflexibilität. Die Module können in nahezu beliebiger Reihenfolge durchlaufen werden.

Die aus der selbststeuernden Prozesskonfiguration resultierende höhere Modellzahl sowie die Einführung zusätzlicher Ereignisse in Form von internen Auslöseereignissen sind nur dann nicht zu rechtfertigen, wenn bei gering komplexen Problemstellungen wenige, geringfügige Unterschiede im Prozessverlauf abzubilden sind.

Es sei darauf hingewiesen, dass eine Substitution der Ereignisdiagramme durch entsprechende EPK, in denen die Bereitstellungs- und Auslöseereignisse durch Verknüpfungsfunktionen miteinander verbunden werden, möglich ist. Die Verwendung eines eigenen Modelltyps (Ereignisdiagramm) für die Verknüpfung der Module führt jedoch zu einer weitaus höheren Transparenz: die Module werden durch EPK, der Prozessverlauf wird durch Ereignisdiagramme modelliert. Zudem wird die Verknüpfungslogik nur in den Ereignisdiagrammen explizit abgebildet: sie können ohne zusätzliches Wissen über die Semantik der Verknüpfungsfunktionen interpretiert und somit maschinell ausgewertet werden.

#### **4.2 Variable Funktionsreihenfolgen**

Die Forderung nach Modul-Intraflexibilität lässt sich verschiedenartig konkretisieren. So können Umwelteinflüsse es erforderlich machen

- Funktionen bzw. Prozessschritte zusammenzufassen,
- einzelne Funktionen bzw. Prozessschritte nicht auszuführen,
- Funktionen bzw. Prozessschritte in unterschiedlicher Reihenfolge durchzuführen,
- auf Fehlersituationen unterschiedlich zu reagieren,
- die möglichen Ergebnisse einer Funktion bzw. eines Prozessschrittes situationsabhängig einzuschränken oder
- Funktionen bzw. Prozessschritte unterschiedlich zu parametrisieren

In diesem Abschnitt wollen wir uns der Thematik variabler Funktionsreihenfolgen zuwenden, welche in ihren einzelnen Ausprägungen schon Gegenstand zahlreicher Arbeiten gewesen ist.

Kiepuszewski behandelt in [Ki02] allgemeine Aspekte des „Interleaved Parallel Routing“. Gegenstand dieser Thematik ist die sequentielle Ausführung von Funktionen in einer Reihenfolge, die erst zum Zeitpunkt der Prozessausführung determiniert wird. Unter besonderer (aber nicht ausschließlicher) Berücksichtigung der Umsetzung in Petri-Netzen wird dieselbe Thematik ebenfalls in [Aa02] behandelt. In [MNN05] wird ein Ansatz zur Realisierung dieses Modellierungsmusters mit Hilfe von EPK vorgestellt.



Einfacher und in ihrer Ausrichtung ähnlich sind die Konzepte in [ST05] und [Gr05]. Während Scheer und Thomas einen neuen Operator einführen, den Sequenzoperator, kommt Grief in seiner Arbeit ohne zusätzliches Konstrukt aus. Er verwendet eine allgemeine Funktion (welche situationsbedingt spezialisiert werden kann) „nächste Aktion bestimmen“ in Kombination mit einem XOR-Verteiler und einer Schleife. Durch den XOR-Verteiler wird explizit gemacht, dass die zur Auswahl stehenden Prozesspfade alternativ durchlaufen werden, und eine Parallelität aus Prozesssicht nicht erlaubt ist. Durch die Schleife wird die Entscheidungsfunktion „Nächste Aktion bestimmen“ (oder eine prozess- bzw. situationsabhängig spezialisierte Funktion) wiederholt durchlaufen, wodurch eine dynamische Entscheidung bezüglich des nächsten auszuführenden Prozesspfades zur Laufzeit des Prozesses ermöglicht wird. Dabei kann auch entschieden werden, dass situationsbedingt nicht alle Prozesspfade durchlaufen werden. Ein konkretes Beispiel zur Anwendung dieses Modellierungsmusters ist in [Gr05, S. 202 - 205] zu finden.

## 5 Gesamtkonstruktion

Während mit der selbststeuernden Prozesskonfiguration Modul-Interflexibilität erreicht wird, besteht die Zielsetzung variabler Funktionsreihenfolgen in der Modul-Intraflexibilität. Für eine Zusammenführung der beiden Ansätze ist die Wahl der Betrachtungsebene, „intermodular“ oder „intramodular“ zu klären: Kann also eine vorliegende Flexibilitätsproblematik prozessbezogen durch ein Modul oder nur durch mehrere Module abgebildet werden.

Die selbststeuernde Prozesskonfiguration mit Hilfe von Ereignisdiagrammen setzt voraus, dass die Umwelteinflüsse, welche die nächste auszuführende Funktion bzw. den nächsten Prozessschritt determinieren, explizit durch Ereignisse benannt und abgebildet werden können. Die in [ST05] und [Gr05] beschriebene variable Funktionsreihenfolge zur Erreichung von Modul-Intraflexibilität kann auch dann bereits angewendet werden, wenn die alternativen Prozesspfade bekannt sind, jedoch noch bezüglich der Bedingungen, unter denen der nächste auszuführende Prozesspfad auszuführen ist, Unsicherheit herrscht.

Der Einsatz der selbststeuernden Prozesskonfiguration bietet sich „Top-down“ folglich an, wenn in einer Flexibilität fordernden dynamischen Umwelt die Gesamtsituation durch Ereignisdiagramme höher aggregiert beschreibbar ist. Einzelne, nicht vollständig durch Ereignisse erfassbare Unsicherheiten der Gesamtsituation können durch variable Funktionsreihenfolgen in Modulen detailliert abgebildet werden.

„Bottom-up“, von einem flexiblen Modul (zur Bewältigung einer Teilproblematik) zu einer flexiblen Modulmenge (für die Gesamtproblematik), kann gegangen werden, wenn sich zunehmend Unsicherheiten durch klar definierte Ereignisse darstellen lassen.

Grundsätzlich denkbar ist aber auch, beide Ansätze (selbststeuernde Prozesskonfiguration und variable Funktionsreihenfolgen) auf einer Betrachtungsebene anzuwenden. Dies sei anhand des bereits bekannten Beispiels zur Auftragsbearbeitung veranschaulicht:

Zu Beginn der Prozessmodellierung ist bekannt, dass unter – noch nicht geklärten Voraussetzungen – ein Auftrag zunächst freizugeben ist, bevor er abgewickelt werden darf. Damit sind die Voraussetzungen zur Abbildung der Prozesskonfiguration in einem Ereignisdiagramm noch nicht gegeben. Die in [ST05] und [Gr05] beschriebene Methode zur Abbildung variabler Funktionsreihenfolgen kann jedoch bereits angewandt werden. Herrscht später Klarheit über die konkreten Bedingungen für die Notwendigkeit einer Freigabe, so sind die Voraussetzungen für eine selbststeuernde Prozesskonfiguration erfüllt.

## Literaturverzeichnis

- [Aa02] van der Aalst, W.M.P.: Workflow Patterns, QUT Technical Report. Queensland University of Technology, Brisbane, 2002.
- [Al98] Allweyer, T.: Adaptive Geschäftsprozesse. Gabler, Wiesbaden, 1998.
- [AS95] Allweyer, T, Scheer, A.-W.: Modellierung und Gestaltung adaptiver Geschäftsprozesse. Institut für Wirtschaftsinformatik der Universität des Saarlandes, Heft 115, 1995.
- [AS04] Aier, S., Schönherr, M.: Enterprise Application Integration als Enabler flexibler Unternehmensarchitekturen. In (Hasselbring, W., Reichert M., Hrsg.): EAI 2004 – Enterprise Application Integration. Tagungsband des GI-/GMDS-Workshops EAI'04, OF-FIS, Oldenburg, 12. – 13.Februar 2004.
- [Ba91] Barney, J.: Firm Resources and Sustained Competitive Advantage. In: Journal of Management, 17 (1991) Nr. 1, S. 99 – 120.
- [Bi93] Bibel, W.: Wissensrepräsentation und Inferenz. Vieweg, Wiesbaden, 1993.
- [BKR05] Becker, J., Kugeler, M., Rosemann, M. (Hrsg.): Prozessmanagement. 5. Auflage. Springer, Berlin, Heidelberg, New York, 2005.
- [BMH05] Beimborn, D., Martin, S. F., Homann, U.: Capability-oriented Modeling of the Firm. In: Proceedings of the IPSI 2005 Conference; Amalfi/Italien (ohne Seitenangaben).
- [Co01] Cockburn, A.: Writing Effective Use Cases. Addison-Wesley, 2001
- [CS94] Chen, R., Scheer, A.-W.: Modellierung von Prozeßketten mittels Petri-Netz-Theorie. Institut für Wirtschaftsinformatik der Universität des Saarlandes, Heft 107, 1994.
- [Da99] Dandl, J.: Objektorientierte Prozeßmodellierung mit der UML und EPK (Arbeitspapiere WI Nr. 12/1999, Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, Univ.-Prof. Dr. Herbert Kargl). Universität Mainz, 1999.
- [Da03] Davis, R.: Business Process Modelling with ARIS. Springer, 2003.
- [Ec2005] Economist Intelligence Unit (Hrsg.): Business 2010 – Embracing the challenge of change. [http://www.eiu.com/site\\_info.asp?info\\_name=eiu\\_SAP\\_business2010](http://www.eiu.com/site_info.asp?info_name=eiu_SAP_business2010) (28.09.2005).
- [Fe86] Feiland, F.-M.: Strategien erfolgreicher mittelständischer Unternehmen (Schriften zur Mittelstandsforschung Nr. 42-1986). C.E. Poeschel, Stuttgart, 1986.
- [GFH90] Gottlob, G, Frühwirth, T, Horn, W.: Expertensysteme. Springer, Wien, New York, 1990.
- [Gr05] Grief, J.: ARIS in IT-Projekten. Vieweg, Wiesbaden, 2005.
- [GSW98] Glaser, H., Schröder, E. F., Werder, A. v. (Hrsg.): Organisationen im Wandel der Märkte. Gabler, Wiesbaden, 1998.
- [HC94] Hammer, M., Champy, J.: Business Reengineering. Campus, Frankfurt/M., New York. 1994.
- [Jo05] Jost, W.: „Der Ball muss ins Tor“. In: SCHEER Magazin, 14 (2005) Nr. 3, S. 6 – 9.

- [Ki02] Kiepuszewski, B.: Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows, PhD thesis, Queensland University of Technology, Brisbane, Australia, 2002.
- [KK92] Kieser, A., Kubicek, H.: Organisation. 3. Auflage. de Gruyter, Berlin, 1992.
- [KNS92] Keller, G., Nüttgens, M., Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage „Ereignisgesteuerter Prozeßketten (EPK)“. Institut für Wirtschaftsinformatik der Universität des Saarlandes, Heft 89, 1992.
- [LA98] Loos, P., Allweyer, T.: Process Orientation and Object-Orientation – An Approach for Integrating UML and Event-Driven Process Chains (EPC) (Publication of the Institut für Wirtschaftsinformatik, University of Saarland, Saarbrücken, Paper 144). Saarbrücken, 1998.
- [MNN05] Mendling, J.; Neumann, G.; Nüttgens, M.: Towards Workflow Pattern Support of Event-Driven Process Chains (EPC). Second GI-Workshop XML4BPM, Karlsruhe 2005.
- [Oe2003] Oestereich, B., Weiss, C., Schröder, C., Weikiens, T., Lenhard, A.: Objektorientierte Geschäftsprozessmodellierung mit der UML. Dpunkt.verlag, Heidelberg, 2003.
- [PDF05] Picot, A., Dietl, H., Franck, E.: Organisation. 4. Auflage. Schäffer-Poeschel, Stuttgart, 2005.
- [Ru99] Rump, F., J.: Geschäftsprozeßmanagement auf der Basis ereignisgesteuerter Prozeßketten. Teubner, 1999.
- [Sc01] Scheer, A.-W.: ARIS - Modellierungsmethoden, Metamodelle, Anwendungen, 4. Auflage, Springer, Berlin usw., 2001.
- [Sc03] Schreyögg, G.: Organisation. 4. Auflage. Gabler, Wiesbaden, 2003.
- [Se05] Seidlmeier, H.: Informationssysteme und Unternehmensprozesse als wettbewerbskritische Ressourcenbündel (in Vorbereitung).
- [SK03] Siedersleben, J., Kurpjuweit, S.: Systemübergreifende Software-Architektur: Erfahrungen und Thesen. In: (Sinz, E. J., Plaha, M., Neckel, P. Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003. Proceedings der Tagung MobIS, Bamberg, 9. – 10. Oktober 2003.
- [SNZ97] Scheer, A.-W., Nüttgens, M., Zimmermann, V.: Objektorientierte Ereignisgesteuerte Prozeßkette (oEPK) – Methode und Anwendung (Veröffentlichungen des Instituts für Wirtschaftsinformatik (IW), Universität des Saarlandes, Heft 141). Saarbrücken, 1997.
- [ST05] Scheer, A.-W., Thomas, O.: Geschäftsprozessmodellierung mit der Ereignisgesteuerten Prozesskette. In: Das Wirtschaftsstudium 34 (2005) Nr. 8-9, S. 1069-1078.
- [Ut97] Uthmann, C.: Nutzenpotenziale der Petrinetztheorie für die Erweiterung der Anwendbarkeit Ereignisgesteuerter Prozeßketten. Vortrag im Rahmen des Workshops an der Universität Oldenburg, 1997.
- [Zi99] Zimmermann, V.: Objektorientiertes Geschäftsprozessmanagement. Deutscher Universitäts-Verlag, Wiesbaden, 1999.