Software-Spezifikation mit ereignisgesteuerten Prozeßketten (EPK)

Constantin Szallies
Energotec Systeme + Automation GmbH, pdv Unternehmensgruppe
szallies@energotec.de

Düsseldorf, 20. Jan 1998

Zusammenfassung

Ereignisgesteuerte Prozeßketten können nicht nur zur Geschäftsprozeßmodellierung, sondern auch zur Spezifikation von Software eingesetzt werden. Dadurch erhält man die Vorteile von umgangssprachlichen und die von formalen Spezifikationsmethoden. Die erstellten Modelle können nach Projektabschluß weiterverwendet werden.
Schlüsselwörter: Ereignisgesteuerte Prozeßkette, Software-Spezifikation, Geschäftsprozeßmodellierung, ARIS-Toolset

1 Einleitung

Im Rahmen des Trends, den Softwareentwicklungsprozeß stärker zu formalisieren, werden verschiedene Ansätze zur formalen Spezifikation von Softwaresystemen entwickelt und eingesetzt. Dieser Artikel versucht Erfahrungen wiederzugeben, die beim Einsatz von ereignisgesteuerten Prozeßketten (EPK) zur Spezifikation von Software gemacht wurden. Es handelte sich um Client/Server Systeme mit graphischer Oberfläche.

EPK ist eine Modellierungsmethode, die ursprünglich aus dem Geschäftsprozeßmodellierungs/Reengineering Umfeld kommt und zur Beschreibung der Ablauforganisation im Unternehmen dient [8]. Wichtig ist dabei die ganzheitliche Sicht, es werden auch Abläufe modelliert, die nicht notwendigerweise EDV-gestützt durchgeführt werden. Andere Anwendungsgebiete für EPK sind:

- Erstellung von *Softwarereferenzmodellen* zum Abgleich von Anforderungen und Ist-Funktionalität.
- Erstellung von *Branchenreferenzmodellen* zur Darstellung von typischen Abläufen einer Branche.
- Erstellung von *Vorgehensmodellen*, z.B. zur Dokumentation des Softwareentwicklungsprozesses.
- *Workflow-Systeme*, die EPK Modelle ausführen können.

Die EPK Modellierung wurde mit dem kommerziellen Produkt ARIS Toolset der Firma IDS [2] durchgeführt. Dieses Produkt bietet auch eine Reihe weiterer Modellierungsmethoden wie z.B. ER-Diagramme zur Datenmodellierung.

Nach einer kurzen Einführung in die Modellierungsmethode erfolgt in Kapitel 3 eine Einordnung der Spezifikation in den Softwareentwicklungsprozeß. Dadurch werden gewünschte Eigenschaften von Software-Spezifikationen definiert. Es wird untersucht, inwiefern EPK Modelle helfen können, diese Eigenschaften zu verbessern. Kapitel 4 geht auf Aspekte ein, die spezifisch für die Erstellung von Software-Spezifikationen sind. Im letzten Kapitel 5 wird ein Zusammenhang zu Geschäftsprozeßmodellen aufgezeigt.

2 Ereignisgesteuerte Prozeßketten

Eine ereignisgesteuerte Prozeßkette ist ein gerichteter Graph mit drei Knotentypen:

- Funktion
- Ereignis
- Logischer Konnektor

Der Prozeß wird als eine Kette von Funktionen modelliert, wobei Funktionen durch Ereignisse verknüpft sind. Eine Funktion hat als Eingang eine Menge von Ereignissen, die eingetreten sein müssen, bevor die Funktion ausgeführt werden kann (Startereignisse). Eine Funktion, die ausgeführt wird, erzeugt wiederum ein oder mehrere Endereignisse, die wiederum als Startereignisse anderer Funktionen dienen können. Durch die-

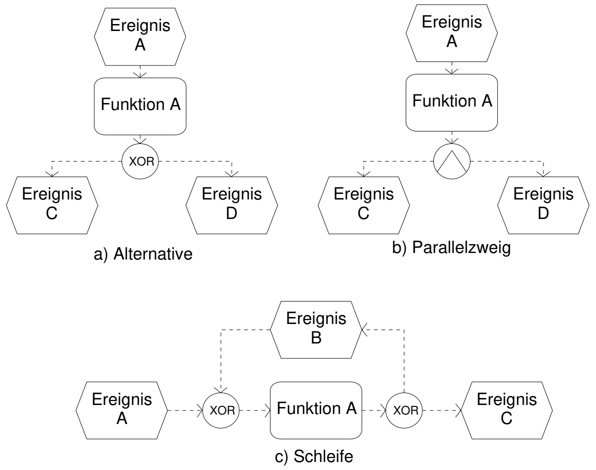

Abbildung 1: Ereignisgesteuerte Prozeßketten

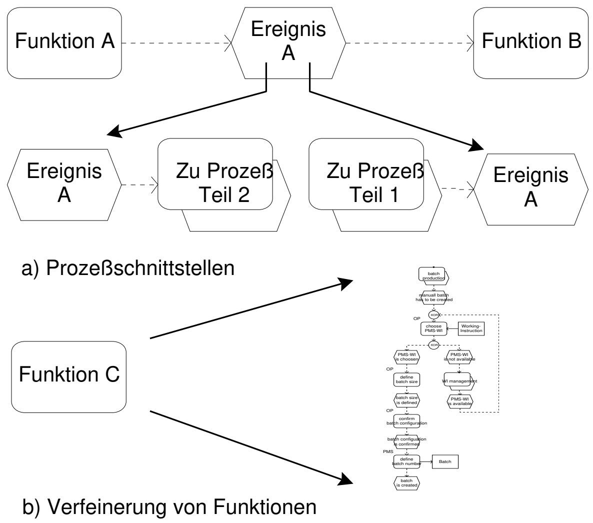

Abbildung 2: Partitionierung von Prozeßmodellen

se Verkettung wird der Kontrollfluß zwischen den Funktionen modelliert.

Durch eine zusätzliche Verknüpfung von Ereignissen durch logische Konnektoren können Schleifen (Bild 1 a) und alternative/parallele Abläufe (Bild 1 b+c) modelliert werden.

Bild 4 und 5 zeigen vollständige EPK Modelle. Das Modell in Bild 4 enthält zwei Teilprozesse, die parallel ausgeführt werden. Der erste AND Konnektor startet und der zweite synchronisiert die parallele Ausführung. Der linke Teilprozeß wiederum enthält einen alternativen Ablauf, der durch das XOR Konnektorpaar definiert wird.

Ereignisse, die keine erzeugende (konsumierende) Funktion besitzen, werden Startereignisse (Endeereignisse) des Prozesses genannt. Beispielsweise ist das Ereignis ”Auftrag liegt vor” das Startereignis der EPK in Bild 4.

Zur Partitionierung von Modellen stehen zwei Modellierungselemente zur Verfügung:

Durch eine *vertikale Partitionierung* werden große Modelle in mehrere Teilmodelle zerschnitten. Der Schnitt erfolgt durch Ereignisse und wird durch sog. Prozeßschnittstellen symbolisiert (Bild 2 a).

Durch eine *horizontale Partitionierung* wird eine Funktion durch ein EPK Modell verfeinert. Die Start- und Endereignisse der Funktion und des Prozesses müssen zueinander passen (Bild 2 b).

In [6] findet man eine Beschreibung der Modellierungsmethode. Eine Literaturliste zum Thema EPK befindet sich online unter http://www.iwi.uni-sb.de/nuettgens/EPK/epk.htm.

3 Die Spezifikation im Softwareentwicklungsprozeß

Die meisten Softwareentwicklungsprozesse folgen einer mehr oder weniger stark modifizierten Version des phasen-orientierten Wasserfallmodells. Ein solcher Standardprozeß ist z.B. das V-Modell [9].

In diesem Modell liegt die Spezifikation eines Softwaresystems zeitlich vor der Implementierung und dient der eindeutigen Festlegung der funktionalen und nicht-funktionalen Eigenschaften.

Die Spezifikation hat im Prozeß zwei Hauptaufgaben:

1. Nach “oben“ dient sie zur Kommunikation mit der Fachseite und als Vertragsgrundlage.
2. Nach “unten“ dient sie
   (a) als Basis zur Erstellung des Software-Designs.
   (b) als Basis zur Erstellung des Testplans.

Die beschriebene Einordnung in den Entwicklungsprozeß liefert die gewünschten Eigenschaften einer Spezifikation. Im folgenden wird untersucht, ob die EPK Modellierung helfen kann, diese Eigenschaften positiv zu beeinflussen.

3.1 Kommunikation mit der Fachseite

Aus der Sicht der Fachseite beschreibt die Spezifikation die Teile der zukünftigen Ablauforganisation, die EDV-unterstützt durchgeführt werden sollen. Die Spezifikation wird in Zusammenarbeit mit der Fachseite erstellt und von ihr verifiziert. Es ist

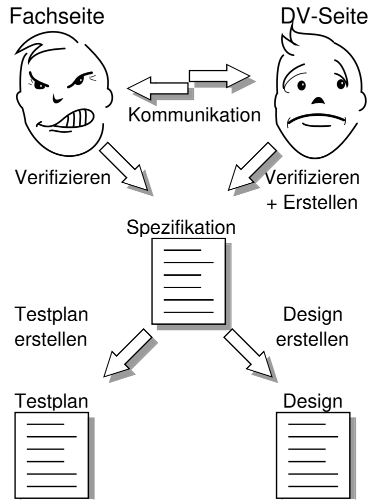

Abbildung 3: Die Spezifikation im Softwareentwicklungsprozeß

daher essenziell, daß die Fachseite die Spezifikation bis ins Detail verstehen kann.

Da die Fachseite normalerweise keinen starken EDV Bezug besitzt, sollte die Spezifikation möglichst wenig formal und leicht zu lesen sein. Ist der Kunde in der Lage ein EPK Modell zu verifizieren?

Da EPK Modellierung eine verbreitete Modellierungsmethode aus der Fachseite ist, ist sie möglicherweise bereits bekannt. Dadurch wird das Verifizieren der Spezifikation vereinfacht, da auf bereits eingeführte Konzepte zurückgegriffen wird.

Hat der Kunde seine Geschäftsprozesse bereits mit EPK Modellen ausgearbeitet, vereinfacht dies die Abbildung der Anforderungen auf die Spezifikation signifikant.

Aus meiner Sicht liegen hier die größten Vorteile der beschriebenen Vorgehensweise.

3.2 Erstellung des Softwaredesigns

Die Spezifikation dient in der nächsten Phase des Entwicklungsprozesses als Basis zur Erstellung des Softwaredesigns. Das Design wird normalerweise nicht in Abstimmung mit der Fachseite erstellt. Eine starke Formalisierung der Spezifikation fördert ihre eindeutige und fehlerfreie Abbildung auf das Design.

Eine Spezifikation in Textform ist zwar leichter zu verstehen, hat aber den Nachteil, daß Vollständigkeit und Fehlerfreiheit schwerer zu verifizieren ist. Zudem neigen solche Dokumente dazu, sehr umfangreich zu werden, was ihre Verständlichkeit wiederum reduziert.

Eine stark formale Spezifikation auf der anderen Seite ist gewöhnlich besser (möglicherweise maschinell) verifizierbar und sehr kompakt. Durch einen schwächeren EDV Bezug ist die Fachseite normalerweise nicht in der Lage, eine stark formale Spezifikation zu verstehen.

Die beschriebene Vorgehensweise ist daher als Kompromiß zwischen den beiden wiedersprüchlichen Anforderungen der formalen Strenge und der Verständlichkeit zu sehen.

Aus der Sicht des Softwareentwicklers fehlt der EPK Methode allerdings eine formale Grundlage. Um dieses Defizit auszugleichen, wurden Ansätze entwickelt, EPK auf Petri-Netze abzubilden [3, 7]. Durch diese Abbildung wird einem EPK Modell eine eindeutige Semantik gegeben, die wiederum als Basis zur maschinellen Verifikation dienen kann. In der Praxis wird dieser Vorteil allerdings durch die hohe Redundanz der EPK Modelle und durch den Anwendungsbezug relativiert, da ein formaler Fehler normalerweise als solcher zu erkennen ist und nicht zu einer Mehrdeutigkeit führt.

3.3 Erstellung der Testpläne

Im Rahmen der Qualitätssicherung wird das erzeugte Softwareprodukt getestet. Die Spezifikation dient als Basis zur Erstellung eines Testplans, in dem, neben der funktionalen Korrektheit, getestet wird, ob alle in der Spezifikation beschriebenen Funktionalitäten implementiert wurden (und nur diese). Der Testplan beschreibt die Durchführung eines White-Box Tests und wird unabhängig von den in der Implementierung involvierten Personen erstellt und durchgeführt.

Die Strukturierung der Testpläne kann der Strukturierung der EPK Modelle folgen. Eine grobe Überprüfung der Abdeckung der Spezifikation kann durchgeführt werden, indem überprüft wird, ob alle Funktionen der EPK Modelle abgedeckt sind.

EPK Modelle können auch zur Beschreibung von System-Schnittstellen herangezogen werden. Es können Modelle erstellt werden, die nicht nur die Transfer-Datenstrukturen beschreiben, sondern auch die konsumierenden bzw. erzeugenden Systemfunktionen beider Systeme. Durch diese Vorgehensweise kann man die Funktionalität des Gesamt-Systems überblicken und so mögliche Design-Fehler besser lokalisieren. Außerdem dient das Modell als gute Basis zur Erstellung der Schnittstellen-Testpläne.

4 Modellierungsaspekte

Dieser Abschnitt beschreibt Erfahrungen, die speziell bei der Spezifikation von Software mittels EPK

Modellen gemacht wurde.

Das System wird durch Modellierung des Systemverhaltens aus Sicht des Benutzers spezifiziert. Es erfolgt normalerweise keine Unterscheidung, ob die Funktionalität verteilt oder zentral ausgeführt wird. Dies ist nicht der Fall, wenn Interaktion zwischen verschiedenen Systemen modelliert werden soll (z.B. EDI-basierte Systeme [5] oder allgemeiner System-Schnittstellen).

4.1 Integration anderer Modelle

Neben den Ablaufmodellen benötigt die Spezifikation natürlich noch andere Modelltypen, wie beispielsweise:

- Datenmodell
- Zustandsautomaten
- Benutzeroberflächen-Prototypen

Mit dem ARIS-Toolset lassen sich ebenfalls Daten- und Organigramme [1] erstellen. Es können Referenzen zwischen den verschiedenen Modelltypen angelegt werden.

Es sollte ferner für jede Funktion angegeben werden, ob sie vom System automatisch ausgeführt wird oder ob Benutzerinteraktionen notwendig sind (manuelle Ausführung). Dies kann man erreichen, indem man im ARIS Toolset Funktionen durch einen Buchstaben kennzeichnet.

Ebenfalls benötigt man für jedes EPK Modell eine Beschreibung in Textform. Sie dient zum leichteren Verständnis und zur Beschreibung von Systemaspekten, die nicht modelliert wurden.

4.2 Strukturierung der Modelle

Da eine vollständige Ablauf-Spezifikation aus zahlreichen EPK Modellen besteht, ist eine geeignete Strukturierung essenziell.

Als geeignet hat sich eine funktionsorientierte Strukturierung der Modelle gezeigt. Ein EPK Modell modelliert dann eine in sich abgeschlossene Systemfunktion. Vorbedingungen, die vor der Ausführung erfüllt sein müssen, werden ebenfalls modelliert.

Einzelne Systemfunktionen können zu Funktionsgruppen zusammengefaßt werden. Das ARIS-Toolset unterstützt diese Vorgehensweise durch eine hierarchische Ablage von Modellen und einem Modelltyp “Funktionsbaum“ [1].

4.3 Horizontale Partitionierung

Eine horizontale Partitionierung bezeichnet die Verfeinerung einer einzelnen Funktion zu einem Prozeßmodell. Die so entstehende Verfeinerungshierachie bildet einen gerichteten, kreisfreien Graphen.

Die horizontale Partitionierung dient folgenden Zielen:

- Reduzierung der Komplexität eines einzelnen Modells
- Um die Lernkurve weniger steil werden zu lassen, können Modelle auf eine Grob- und eine Fein-Ebene aufgeteilt werden. In der Grob-Ebene bekommt man eine schnelle Übersicht über das System und die Fein-Ebene besteht aus verfeinerten Funktionen der Grob-Ebene.
- Mehrfache Verwendung von Prozeßmodellen.

4.4 Vertikale Partitionierung

Durch die vertikale Partitionierung wird eine große Kette in mehrere kleine zerlegt, um sie beispielsweise besser in ein Papierdokument einbetten zu können.

Die Zerlegung ist so vorzunehmen, daß möglichst keine Schleifen zwischen den verschiedenen Teilen verlaufen.

4.5 Granularität

Bei der Modellierung der Abläufe stellt sich die Frage, wie genau diese modelliert werden sollen. Einerseits möchte man eine möglichst genaue Modellierung um Mehrdeutigkeiten auszuschließen, andererseits möchte man die Modelle klein halten, um Übersichtlichkeit zu gewährleisten.

Es ist zu beachten, daß alle Modelle einer Ebene ungefähr gleich granular sind. Bestimmte Aspekte sollten entweder in allen oder in keinem Modell berücksichtigt werden.

4.6 Verkleinerung von EPK Modellen

Man kann versuchen EPK Modelle zur Reduzierung der Komplexität durch Modellierungskonventionen verkleinern. Dies ist auch wichtig, da EPK Modelle im Vergleich zu äquivalenten Modellen anderer Methoden mehr Knoten aufweisen (z.B. FUNSOFT-Netze [4]).

Eine Strategie ist es, gewisse Teilabläufe einfach nicht zu spezifizieren. Dies geschieht beispielsweise, indem ein durch einen Fehler verursachter Rücksprung nicht als Schleife sondern als Endereignis modelliert wird. Das Systemverhalten nach Eintreten des Ereignisses wird dann in Textform beschrieben. Damit erhält das Ereignis allerdings eine andere Semantik und der Prozeß verliert die sog. “Liveness“ [7] Eigenschaft.

Eine andere mögliche Modellierungskonvention betrifft die Spezifikation von Eingabemasken. Die Modellierung jedes Eingabefeldes als Funktion (z.B. "Datum eingeben") erweist sich oft als zu aufwendig. Oft reicht es aus, die komplette Eingabemaske als eine Funktion zu modellieren, die auf das Datenmodell und/oder einen Benutzeroberflächen-Prototypen verweist.

4.7 Parallele Teilprozesse

Im EPK Modell wird eine parallele Ausführung von Teilprozessen durch ein AND Konnektorpaar modelliert (wie in Bild 4). Die Erfahrung hat gezeigt, daß dieses Konstrukt in Softwarespezifikationsmodellen seltener auftritt. Dies hat unter anderem folgende Gründe:

- Eine Menge von Teilprozessen, die unabhängig voneinander ausgeführt werden könnten, sollen in der Software aus Gründen der Benutzerfreundlichkeit und Effizienz von einer Person in Sequenz ausgeführt werden. Die Reihenfolge soll in der Spezifikation festgeschrieben werden.
- Teilprozesse, die in einem Geschäftsprozeßmodell parallel modelliert werden können, können gar nicht unabhängig voneinander ausgeführt werden. Dies ist z.B. dann der Fall, wenn der Abbruch eines Teilprozesses die Bearbeitung der anderen Teilprozesse überflüssig macht.

4.8 Schwer-modellierbare Zusammenhänge

Manchmal stellt sich während der Modellierung heraus, daß ein relevanter Sachverhalt mit EPK Modellen nur schwer oder unzureichend dargestellt werden kann. So ist es z.B. schwierig, in einem einzelnen Modell die Erzeugung und Synchronisation einer variablen Anzahl von Teilprozessen zu modellieren. Asynchron auftretende Ereignisse können nur als Startereignisse eines separaten Prozeßmodells aufgefangen werden.

Dadurch wird man manchmal gezwungen, Sachverhalte, die logisch in ein Prozeßmodell gehören, in mehrere Modelle aufzuteilen. Der Zusammenhang der Modelle wird dann in Textform beschrieben.

Es sei beispielsweise folgender Sachverhalt gegeben: Ein Unternehmen möchte einen Auftrag ausführen, indem er diesen in n Unteraufträge zerlegt. Nach Ausführung aller Unteraufträge wird der Auftrag fertig gemeldet.

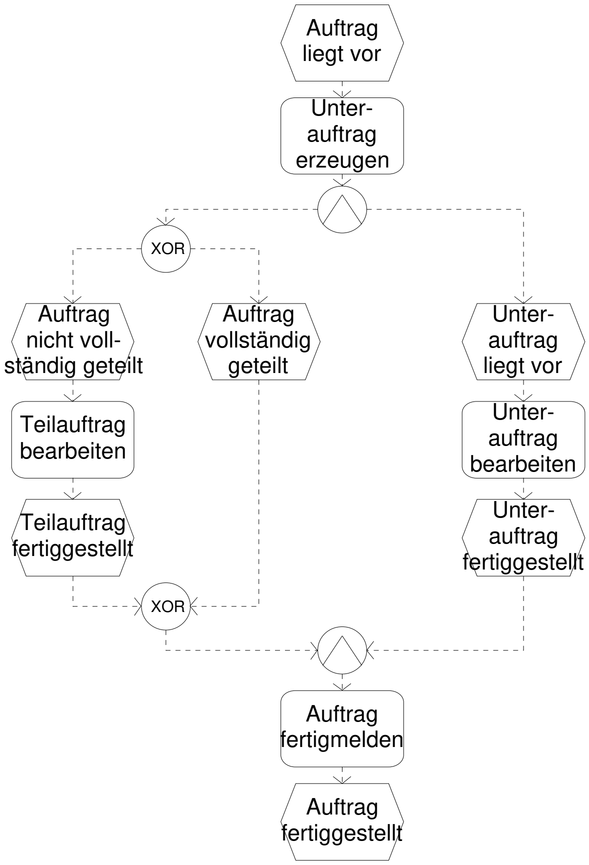

Abbildung 4: Prozeßmodell "Auftrag bearbeiten": Alternative 1

Bild 4 zeigt eine rekursive Modellierung des Sachverhalts. Der verzweigende XOR Konnektor modelliert eine Fallunterscheidung zwischen $n = 1$ und $n > 1$, wobei im linken Zweig das Problem von n auf $n - 1$ reduziert wird. Die Funktion "Teilauftrag bearbeiten" wird nicht weiter verfeinert.

Das Modell in Bild 5 modelliert ebenfalls den beschriebenen Sachverhalt. Diesmal muß das Modell n mal ausgeführt werden, um den Auftrag fertig zu melden. Wie das "Letzter Unterauftrag fertiggestellt" Ereignis erzeugt wird, bleibt unklar und muß in Textform beschrieben werden.

Wie jetzt die Systemfunktionen "Auftrag stornieren" und "Unterauftrag stornieren" modelliert werden könnten, bleibt dem Leser als kleine Aufgabe überlassen.

5 Integration zu Geschäftsprozeßmodellen

Wie kann die Fachseite ihre Geschäftsprozesse in der Spezifikation wiederfinden und wie kann sie verifizieren, daß die Software die Ablauf- und Auf-

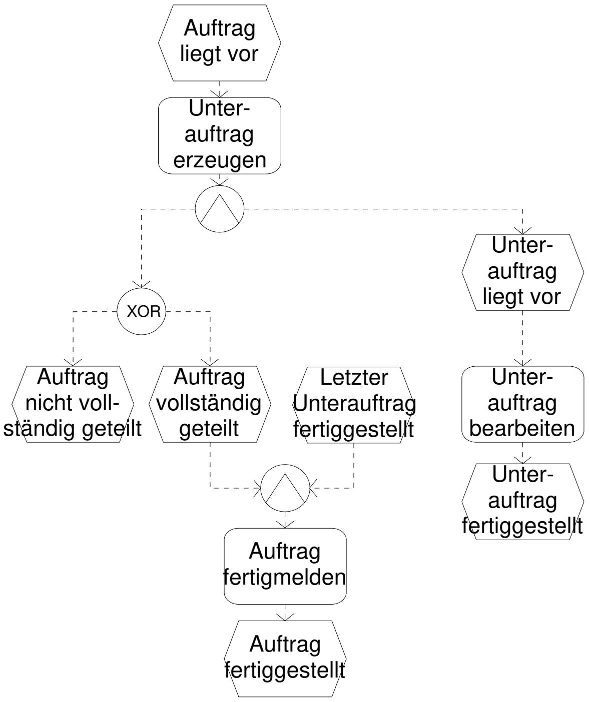

Abbildung 5: Prozeßmodell "Auftrag bearbeiten": Alternative 2

bauorganisation adäquat unterstützt?

Nach Modellierung der Geschäftsprozesse können Teile der Spezifikation integriert werden. Dies geschieht, indem einzelne Modelle des Spezifikation in die Geschäftsprozeßmodelle als Funktionen eingebettet werden, d.h. es werden vorher nicht-zusammenhängende System-Funktionen durch nicht DV-gestützte Abläufe zu einem ganzheitlichen Geschäftsprozeß integriert.

Nach Projektabschluß kann die Spezifikation zu einem Softwarereferenzmodell aufgearbeitet werden. Dadurch werden Abläufe, Funktionalitäten und organisatorische Voraussetzungen des Softwareprodukts dokumentiert. Das Modell kann beispielsweise für einen Vergleich der Ist-Funktionalität und der Anforderungen eines Neukunden verwendet werden.

6 Resüme

Der Einsatz von EPK Modellen in Software-Spezifikationen bietet Vorteile, wenn die Fachseite mit dieser Modellierungsmethode vertraut ist. Liegen außerdem noch relevante Geschäftsprozeßmodelle vor, wird das Verifizieren der Spezifikation nochmals vereinfacht. Die Modellierungsmethode bietet einige Vorteile einer formalen Spezifikation, bleibt aber auch noch für DV-Laien verständlich. Nach Projektabschluß können die Modelle weiterverwendet werden.

Als nachteilig ist die manchmal unzureichende Modellierungsmächtigkeit und die Größe der resultierenden Modelle anzusehen. Dazu kommen einige Schwächen der verwendeten Software. Die Stabilität der Software könnte besser sein. Modelle können nicht komfortabel in jede beliebige Textverarbeitung eingebettet werden. Modelle, die größer als eine Seite sind, werden ungünstig ausgedruckt.

Literatur

[1] ARIS Methodenhandbuch, ARIS-Toolset Dokumentation

[2] http://www.ids-scheer.de

[3] *R. Chen, A.W. Scheer*: Modellierung von Prozeßketten mittels Petri-Netz Theorie, Veröffendlichungen des Instituts für Wirtschaftsinformatik, Heft 107, Saarbrücken 1994

[4] *V. Gruhn, R. Jegelka*: An Evaluation of FUNSOFT Nets, Software Process Technology — Proceedings of the 2nd European Software Process Modelling Workshop, S. 194 - 214, Lecture Notes in Computer Science, 1992

[5] *V. Gruhn*: Elektronischer Datenaustausch in zwischenbetrieblichen Geschäftsprozessen, Wirtschaftsinformatik 39 (1997) 3, S. 255 - 230

[6] *G. Keller, M. Nüttgens, A.W. Scheer*: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)", Veröffendlichungen des Instituts für Wirtschaftsinformatik, Heft 89, Saarbrücken 1992

[7] *P. Langner, C. Schneider, J. Wehler*: Prozeßmodellierung mit ereignisgesteuerten Prozeßketten (EPKs) und Petri-Netzen, Wirtschaftsinformatik 39 (1997) 5, S. 479 - 489, 1997

[8] *A.W. Scheer*: ARIS-House of Business Engineering, Veröffendlichungen des Instituts für Wirtschaftsinformatik, Heft 133, Saarbrücken 1996

[9] *A. Bröhl, W. Dröschel*: Das V-Modell, Oldenburg, 1995