Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK)

Markus Nüttgens

Universität Trier
Wirtschaftsinformatik II
Postfach 3825, D-54286 Trier
E-Mail: markus@nuettgens.de

Frank J. Rump

Fachhochschule OL/Ostfriesland/WHV
Fachbereich Technik
Constantiaplatz 4, D-26723 Emden
E-mail: rump@informatik-emden.de

Abstract: Die Ereignisgesteuerte Prozesskette (EPK) wurde zur Dokumentation von Geschäftsprozessen entwickelt und hat in der Praxis eine weite Verbreitung gefunden. Aufgrund der hohen Akzeptanz und der wachsenden Bedeutung prozessorientierter Organisationsstrukturen dient sie zunehmend als Grundlage für ein integriertes Geschäftsprozessmanagement. Ein durchgängiges Managementkonzept zur werkzeuggestützten Planung, Steuerung, Ausführung und Kontrolle von Geschäftsprozessen erfordert eine korrekte Formalisierung und Implementierung der EPK-Syntax und -Semantik. Die in der Theorie und Praxis dokumentierten Beiträge zur EPK-Formalisierung leisten dies nur mit wesentlichen Einschränkungen. In diesem Beitrag erfolgt eine Formalisierung des Kontrollflusskonzeptes auf der Grundlage der ursprünglichen EPK-Syntax und -Semantik. Der vorliegende Ansatz kann um ein Ressourcen-, Mengen- und Zeitkonzept erweitert werden und bietet Anwendern und Werkzeugherstellern einen stabilen Bezugsrahmen zur korrekten Modellierung und Anwendung von EPK-Geschäftsprozessmodellen.

1 Einführung

Die Ereignisgesteuerte Prozesskette (EPK) wurde am Institut für Wirtschaftsinformatik (IWi) der Universität des Saarlandes in Zusammenarbeit mit der SAP AG zur Dokumentation von Geschäftsprozessen entwickelt [KNS92]. Sie ist zentraler Bestandteil der SAP-Referenzmodelle [Ke99] und der ARIS-Konzepte [Sc99, Sc01] und somit Grundlage modellgetriebener Ansätze für ein durchgängiges und werkzeuggestütztes Geschäftsprozessmanagement.

Ein Managementkonzept zur werkzeuggestützten Planung, Steuerung, Ausführung und Kontrolle von Geschäftsprozessen erfordert eine hinreichende Spezifikation des EPK-Konzeptes. Die Formalisierung vermeidet unterschiedliche Interpretationen und ist Basis für korrekte Anwendungen und Implementierungen von Geschäftsprozessmodellen. In der Literatur sind bislang nur einige wenige Ansätze zur formalen Spezifikation der EPK-Syntax und -Semantik vorgeschlagen worden.

Nüttgens, M.; Rump, F.: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK), in: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen (Promise'2002), Gemeinsames Fachgruppentreffen der GI-Fachgruppen „Petrinetze und verwandte Systemmodelle" und „Entwicklungsmethoden für Informationssysteme und ihre Anwendung" (EMISA), Hasso-Plattner-Institut für Softwaresystemtechnik an der Universität Potsdam, 9.-11. Oktober 2002, Potsdam 2002

Im Regelfall wird eine „Übersetzersemantik“ in Zustandsdiagramme (Petrinetze, Statecharts etc.) zugrunde gelegt, um die vorhandenen formalen Analysetechniken zur Verifikation von (Geschäfts-)Prozessmodellen nutzen zu können. Exemplarische Forschungsansätze finden sich bei Chen/Scheer und Hoffmann/Scheer [CS94, HSH95], Langner/Schneider/Wehler [LSW97a, LSW97b, LSW97c, LSW97d, LSW98], Rodenhagen/Modt [Ro97, MR00], v. Uthmann [Ut97, Ut98], Weikum/Wodtke [We97, Wo97], Volkmer [Vo97], v. d. Aalst [Aa98, Aa99] und Rittgen/Dehnert [Ri99a, Ri99b, Ri99c, Ri99d, Ri00a, Ri00b, Ri00c, De01, DR01]. Weitere eigenständige formale Ansätze zur Syntax- und Semantikdefinition finden sich bei Rump [Ru95, ZR96, Ru97a, Ru97b, Ru97c, Ru99], Moll [Mo96] und Heimig [He01]. Einen konzeptionellen Ansatz im Rahmen der „Grundsätze ordungsgemäßer Modellierung“ verfolgen Becker/ Rosemann/Schütte [BRS95, Ro96].

Keiner dieser Ansätze spezifiziert das EPK-Konzept vollständig auf der Grundlage der ursprünglichen Syntax und Semantik. Vielmehr werden entweder wesentliche Bestandteile der EPK-Syntax und -Semantik nicht oder mit mehr oder weniger starken Einschränkungen formalisiert. Insbesondere die Übersetzersemantiken in Petrinetze führen bei der Formalisierung des Synchronisationsverhaltens bei (X)OR-Verknüpfungsoperatoren zu Problemen, da deren Semantik nicht lokal beschränkt ist. Eine unreflektierte Übernahme der Verifikationskonzepte für Petrinetze führt ebenfalls zu Differenzen bei der Beurteilung ausgewählter, besonders kritischer EPK-Eigenschaften (z.B. Verklemmungsfreiheit von EPKs).

In Abb. 1 ist exemplarisch ein EPK-Schema zur Beschreibung des Geschäftsprozesses „Kreditantrag bearbeiten“ aufgeführt. Das Kontrollflusskonzept wird als eine Abfolge von Ereignissen, Funktionen, Verknüpfungsoperatoren und Prozesswegweisern beschrieben und kann um den Ressourcenaspekt (Organisationseinheiten, Informations- und Sachobjekte) ergänzt werden. Funktionen und Prozesswegweiser können aus pragmatischen Gründen horizontal und vertikal (de-)komponiert werden. Dies wird logisch über die Verwendung gemeinsamer Ereignisse (E8, E9, E10) ausgedrückt. Prozesswegweiser zeigen explizit Schnittstellen zwischen vor- und nachgelagerten (Teil-) Prozessketten an und dienen insbesondere bei großen EPK-Schemata als wichtige Navigationshilfe. Die relevanten Ressourcenaspekte können ebenfalls direkt im Kernmodell oder in hierarchisierten Teilmodellen abgebildet werden.

Die aus dem EPK-Schema resultierenden Prozessszenarien sind weitgehend selbsterklärend: Innerhalb des Kernprozesses „Kreditantrag bearbeiten“ wird der Kreditantrag erfasst (F1) und einer standardisierten Risikoprüfung unterzogen (F2). Im Falle des negativen Ausgangs der (Erst-)Prüfung erfolgt eine weitere fallspezifische Abschätzung (F3). Diese endet entweder mit der Ablehung (F4) oder einer Wiedervorlage zur Risikoprüfung (F2). Im Falle einer positiven Risikoprüfung ist es von Relevanz, ob es sich um einen Neukunden handelt. In diesem Fall erfolgt nebenläufig zur Erstellung des Kreditvertrages (F5) eine Bedarfsanalyse (F6). Sobald der Kreditvertrag vorbereitet ist, kann dieser von den Vertragspartnern unterschrieben werden (F7). Bei Neukunden kann eine Nachberatung (F8) frühestens dann erfolgen, wenn der Kreditvertrag vorbereitet ist und die Bedarfsanalyse abgeschlossen ist.

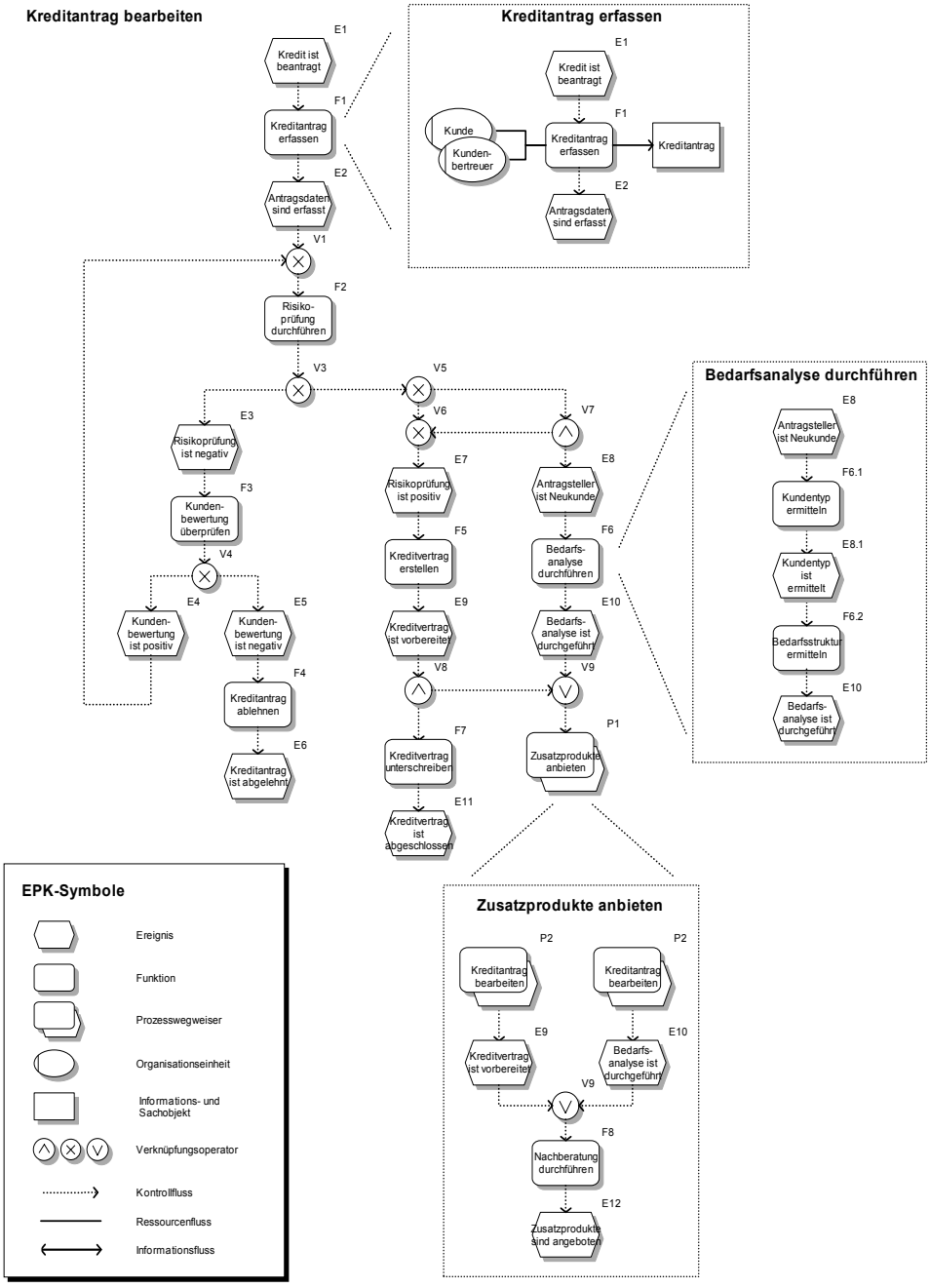

Abb. 1: Ereignisgesteuerte Prozesskette: Fallbeispiel „Kreditantrag“

2 Syntax- und Semantikdefinition

Nachfolgend wird eine Formalisierung des EPK-Kontrollflusskonzeptes durch eine umfassende Syntax- und Semantikdefinition vorgestellt. Hierbei wird ein operationaler Ansatz zur Semantikdefinition verfolgt, da dieser unmittelbar die schrittweise Berechnung sämtlicher erreichbaren Zustände und darauf aufbauende Analysen ermöglicht [Ru99]. Für die Beschreibung der Syntax und Semantik von EPKs werden die Definitionen der Multimenge und der darauf möglichen Operationen benötigt.

Definition 2.1 (Multimenge) Eine Multimenge M über einer Menge A ist ein Paar $M = (A, m)$, wobei m die Abbildung $m : A \to N_0$ ist. $m(a)$ heißt Multiplizität von $a \in A$. A_{MS} stellt die Menge aller Multimengen über A dar. Eine Multimenge kann durch ihre formale Summe dargestellt werden: $M = \sum_{a \in A} m(a)a$.
Seien $M_1 = (A, m_1)$, $M_2 = (A, m_2) \in A_{MS}$ und $a \in A$. Es gilt

$a \in M_1$, falls $m_1(a) \geq 1$	(Elementbeziehung)
$M_1 \leq M_2$, falls $\forall a \in A : m_1(a) \leq m_2(a)$	(Inklusion)
$M_1 = M_2$, falls $M_1 \leq M_2$ und $M_2 \leq M_1$	(Gleichheit)
$M_1 + M_2 = \sum_{a \in A} (m_1(a) + m_2(a))a$	(Addition)
$M_1 - M_2 = \sum_{a \in A} ((m_1(a) - m_2(a)) \max 0)a$	(Subtraktion)
$\mid M_1 \mid = \sum_{a \in A} m_1(a)$	(Kardinalität)

2.1 Syntaxdefinition

Dieser Abschnitt beschreibt die Syntax einer EPK, bei der zunächst nur der Kontrollfluss betrachtet wird und somit als Knoten nur Ereignisse, Funktionen, Verknüpfungsoperatoren und Prozesswegweiser zugelassen werden. Weiterhin beschränkt sich die folgende Definition zunächst auf nicht-hierarchische EPKs.

Definition 2.2 (flaches EPK-Schema) Ein flaches EPK-Schema A ist ein Tupel $A = (E, F, P, V, C, S_0)$, für welches gilt:

- E ist eine Menge von Ereignissen mit $E \neq \emptyset$,
- F ist eine Menge von Funktionen mit $F \neq \emptyset$,
- P ist eine Menge von Prozesswegweisern und
- V ist eine Menge von Verknüpfungsoperatoren, die sich in paarweise disjunkte Teilmengen V_{OR}, V_{XOR} und V_{AND} aufteilt, so dass $V = V_{OR} \cup V_{XOR} \cup V_{AND}$.
- E, F, P und V sind paarweise disjunkt.
- Sei $K = E \cup F \cup P \cup V$. Die Multimenge $C = (K \times K, m_C)$ beschreibt den Kontrollfluss.
- $S_0 \subseteq K_{MS}$ ist eine Menge von Startbelegungen.

Die Menge S_0 stellt in der Regel eine Teilmenge der Startereignisse dar und legt fest, welche Kombinationen von eingetretenen Startereignissen eine initiale EPK-Instanz aufweisen kann. Falls keine Prozesswegweiser im EPK-Schema verwendet werden ($P = \varnothing$), lässt sich das flache EPK-Schema auch kurz durch $A = (E, F, V, C, S_0)$ darstellen.
Zur Vereinfachung werden folgende Schreibweisen eingeführt:

1. $j \rightarrow_C k :\Leftrightarrow (j,k) \in C$
2. $\bullet k = (K, m_{V_k})$, wobei $m_{V_k}(j) = m_C((j,k))$ ist, sei die Multimenge der direkten Vorgänger von k , und
3. $k\bullet = (K, m_{N_k})$, wobei $m_{N_k}(j) = m_C((k,j))$ ist, enthält die direkten Nachfolger von k.
4. $j \rightarrow_C^* k :\Leftrightarrow \exists a_1,...,a_n \in K, n > 1 : j = a_1 \rightarrow_C a_2 \rightarrow_C ... \rightarrow_C a_n = k$; die Relation $\rightarrow_C^*$ beschreibt, welche Elemente von K durch den Kontrollfluss miteinander verbunden sind.
5. $j \rightarrow_C^V k :\Leftrightarrow \exists v_1,...,v_n \in V, n \geq 0 : j \rightarrow_C v_1 \rightarrow_C ... \rightarrow_C v_n \rightarrow_C k$; die Relation $\rightarrow_C^V$ beschreibt, welche Elemente von K im Kontrollfluss nur über Verknüpfungsoperatoren miteinander verbunden sind.

Zusätzlich werden folgende Mengen definiert:

$E_S = \{e \in E \mid \neg\exists g \in E : g \rightarrow_C^* e\}$	Menge der Startereignisse,
$E_E = \{e \in E \mid \neg\exists g \in E : e \rightarrow_C^* g\}$	Menge der Endereignisse,
$S_{AND} = \{v \in V_{AND} \mid \mid \bullet v \mid = 1\}$	Menge der AND-Split-Operatoren,
$S_{XOR} = \{v \in V_{XOR} \mid \mid \bullet v \mid = 1\}$	Menge der XOR-Split-Operatoren,
$S_{OR} = \{v \in V_{OR} \mid \mid \bullet v \mid = 1\}$	Menge der OR-Split-Operatoren,
$J_{AND} = \{v \in V_{AND} \mid \mid \bullet v \mid > 1\}$	Menge der AND-Join-Operatoren,
$J_{XOR} = \{v \in V_{XOR} \mid \mid \bullet v \mid > 1\}$	Menge der XOR-Join-Operatoren,
$J_{OR} = \{v \in V_{OR} \mid \mid \bullet v \mid > 1\}$	Menge der OR-Join-Operatoren.

Ein syntaktisch korrektes, flaches EPK-Schema $A = (E, F, P, V, C, S_0)$ muss weiterhin die folgenden Eigenschaften erfüllen:

1. Sei $C_S = \{(j,k) \mid (j,k) \in C\}$. $G = (K, C_S)$ ist ein gerichteter und zusammenhängender Graph.

2. Alle Kontrollflusskanten haben die Multiplizität 1:[1]
   $\forall (j,k) \in C : m_C((j,k)) = 1$
3. Funktionen besitzen genau eine eingehende und genau eine ausgehende Kontrollflusskante:
   $\forall f \in F : | \bullet f | = | f \bullet | = 1$
4. Ereignisse haben genau eine eingehende und/oder genau eine ausgehende Kontrollflusskante:
   $\forall e \in E : | \bullet e | \leq 1 \wedge | e \bullet | \leq 1$
   (Falls ein Ereignis nur eine Kontrollflusskante aufweist, handelt es sich um ein Start- oder Endereignis: $\forall e \in E : | \bullet e | + | e \bullet | = 1 \Rightarrow e \in E_S \cup E_E$)
5. Prozesswegweiser haben genau eine eingehende oder eine ausgehende Kontrollflusskante:
   $\forall p \in P : | \bullet p | + | p \bullet | = 1$
6. Verknüpfungsoperatoren haben entweder eine eingehende und mehrere ausgehende (Split-Operator) oder mehrere eingehende und eine ausgehende Kontrollflusskante (Join-Operator):
   $\forall v \in V : (| \bullet v | = 1 \wedge | v \bullet | > 1) \vee (| \bullet v | > 1 \wedge | v \bullet | = 1)$
7. Es gibt keinen gerichteten Kreis im EPK-Schema, der nur aus Verknüpfungsoperatoren besteht:
   $\forall u, v \in V : u \rightarrow_C^V v \Rightarrow u \neq v$
8. Ereignisse sind nur mit Funktionen und Prozesswegweisern (möglicherweise über Verknüpfungsoperatoren) verbunden:
   $\forall e \in E, f \in K - V : e \rightarrow_C^V f \Rightarrow f \in F \cup P$
9. Funktionen und Prozesswegweiser sind nur mit Ereignissen (möglicherweise über Verknüpfungsoperatoren) verbunden:
   $\forall f \in F \cup P, e \in K - V : f \rightarrow_C^V e \Rightarrow e \in E$
10. Nach Ereignissen folgt kein XOR- oder OR-Split-Operator im Kontrollfluss:[2]
    $\forall e \in E : e \rightarrow_C^V v \wedge v \in V \Rightarrow v \in S_{AND} \cup J_{OR} \cup J_{XOR} \cup J_{AND}$
11. Es gibt mindestens ein Start- und mindestens ein Endereignis:
    $| E_S | > 0 \wedge | E_E | > 0$
12. In einer Startbelegung dürfen nur Startereignisse und diese nur mit der Multiplizität 1 enthalten sein:
    $\forall S \in S_0 \forall e \in S : e \in E_S \wedge m_S(e) = 1$
13. Jedes Startereignis ist in einer Startbelegung enthalten:
    $\forall e \in E_S \exists S \in S_0 : e \in S$

---

[1] Anschaulich gesprochen wird hierdurch die Einfachheit des Graphen gefordert. Die dadurch mögliche Darstellung der Kanten als Teilmenge von *K x K* wird allerdings nicht gewählt, da die Einfachheit bei den in [Ru99] eingeführten Junktornetzen nicht mehr gegeben ist, aber trotzdem dieselbe Semantikdefinition anwendbar sein soll.

[2] Somit werden nur die in [KN92] dargestellten Verknüpfungsarten erlaubt. Diese Einschränkung entspricht auch der Interpretation des Ereignisbegriffes im Kontext deterministischer endlicher Zustandsautomaten.

Definition 2.3 (syntaktisch korrektes, flaches EPK-Schema) Ein flaches EPK-Schema $A = (E, F, P, V, C, S_0)$ heißt syntaktisch korrekt, wenn das Schema die Eigenschaften 1-13 erfüllt.

Geschäftsprozesse werden in der Praxis aufgrund der besseren Übersichtlichkeit und der Möglichkeit der Wiederverwendung von Prozessteilen nicht durch ein einzelnes EPK-Schema, sondern durch eine Menge von EPK-Schemata beschrieben, die über Prozesswegweiser oder Funktionen miteinander verbunden sind. Diese hierarchischen EPK-Schemata werden im folgenden eingeführt.

Definition 2.4 (hierarchisches EPK-Schema) Sei $E' = \{A_1, ..., A_n\}$ eine Menge von EPK-Schemata. Ein hierarchisches EPK-Schema A ist ein Tupel $A = (E, F, P, V, C, S_0, H)$, für welches

- $A' = (E, F, P, V, C, S_0)$ ein flaches EPK-Schema ist und
- $H \subseteq (F \cup P) \times E'$ die Verknüpfung zu einem anderen EPK-Schema herstellt (Hierarchierelation)

Die Relation H ordnet somit einer Funktion oder einem Prozesswegweiser ein anderes EPK-Schema zu. Eine Funktion, die derart durch ein anderes EPK-Schema verfeinert wird, wird im folgenden hierarchisierte Funktion genannt. In einer EPK-Schemamenge werden nun EPK-Schemata zusammengefasst, bei denen die über Prozesswegweiser oder Funktionen referenzierten EPK-Schemata auch wieder selbst Elemente der EPK-Schemamenge sind.

Definition 2.5 (EPK-Schemamenge) Eine EPK-Schemamenge $E^* = \{A_1, ..., A_n\}$ ist eine Menge von (flachen oder hierarchischen) EPK-Schemata, bei der für alle hierarchischen EPK-Schemata $A_i = (E_i, F_i, P_i, V_i, C_i, S_{0_i}, H_i)$ gilt: $H_i \subseteq (F_i \cup P_i) \times E^*$.

Durch die nachfolgende Definition eines syntaktisch korrekten, hierarchischen EPK-Schemas wird festgelegt, wie die Verknüpfung von EPK-Schemata einer EPK-Schemamenge zu erfolgen hat:

Definition 2.6 (syntaktisch korrektes, hierarchisches EPK-Schema) Sei $E^* = \{A_1, ..., A_n\}$ eine EPK-Schemamenge. $A_i = (E_i, F_i, P_i, V_i, C_i, S_{0_i}, H_i) \in E^*$ ist ein syntaktisch korrektes, hierarchisches EPK-Schema, wenn folgende Forderungen erfüllt sind:

1. $A' = (E_i, F_i, P_i, V_i, C_i, S_{0_i})$ ist ein syntaktisch korrektes, flaches EPK-Schema.
2. Jedem Prozesswegweiser ist über die Hierarchierelation genau ein EPK-Schema zugeordnet:
   $\forall p \in P_i : | \{A \in E^* \mid (p, A) \in H_i\} | = 1$

3. Jeder Funktion wird maximal ein EPK-Schema zugewiesen:
   $\forall f \in F_i : |\{A \in E^* \mid (f, A) \in H_i\}| \leq 1$
4. Die Menge der vorangehenden Ereignisse einer hierarchisierten Funktion entspricht der Menge der Startereignisse des referenzierten EPK-Schemas:
   $\forall f \in F_i : (f, A_j) \in H_i \Rightarrow \{e \in E_i \mid e \rightarrow_C^V f\} = E_{S_j}$
5. Die Menge der nachfolgenden Ereignisse einer hierarchisierten Funktion entspricht der Menge der Endereignisse des referenzierten Schemas:
   $\forall f \in F_i : (f, A_j) \in H_i \Rightarrow \{e \in E_i \mid f \rightarrow_C^V e\} = E_{E_j}$
6. Die Menge der vorangehenden Ereignisse eines Prozesswegweisers ist eine Teilmenge der Startereignisse des referenzierten EPK-Schemas:
   $\forall p \in P_i : (p, A_j) \in H_i \Rightarrow \{e \in E_i \mid e \rightarrow_C^V p\} \subseteq E_{S_j}$
7. Die Menge der nachfolgenden Ereignisse eines Prozesswegweisers ist eine Teilmenge der Endereignisse des referenzierten Schemas:
   $\forall p \in P_i : (p, A_j) \in H_i \Rightarrow \{e \in E_i \mid p \rightarrow_C^V e\} \subseteq E_{E_j}$
8. Das hierarchische EPK-Schema A_i ist nicht über die Hierarchierelation mit sich selbst verbunden (Verbot der Rekursion):
   $\neg\exists A_{i_1}, ..., A_{i_j} \in E^* : (\forall k, 1 \leq k < j \exists f \in F_{i_k} \cup P_{i_k} : (f, A_{i_{k+1}}) \in H_{i_k}) \wedge A_{i_1} = A_{i_j} = A_i$

Diese Forderungen bieten die Möglichkeit, aus einem syntaktisch korrekten, hierarchischen EPK-Schema ein flaches EPK-Schema zu erzeugen („Flachklopfen"), indem hierarchisierte Funktionen bzw. Prozesswegweiser durch die in der Hierarchierelation referenzierten EPK-Schemata sukzessive verfeinert werden, wobei die Verknüpfung durch die in beiden Schemata enthaltenen, benachbarten Ereignisse spezifiziert ist. Aufgrund dieser Möglichkeit werden zur Semantikdefinition nur flache Schemata ohne Prozesswegweiser betrachtet, was entsprechend dieser Ausführungen keine Einschränkung darstellt.

2.2 Semantikdefinition

Zur Definition der Semantik wird zunächst das flache EPK-Schema in ein Vorverknüpfer-ergänztes, flaches EPK-Schema umgewandelt. Dazu wird vor jedem Join-Operator für jede Kante ein Vorverknüpfer in den Kontrollfluss eingefügt, um feststellen zu können, welche eingehenden Pfade aktiviert sind.

Definition 2.7 (Vorverknüpfer-ergänztes, flaches EPK-Schema) Ein Vorverknüpfer-ergänztes EPK-Schema $A' = (E, F, V, W, C', S_0)$ zu einem syntaktisch korrekten EPK-Schema $A = (E, F, V, C, S_0)$ ergibt sich durch folgende Ergänzung einer Menge von Vorverknüpfern W im Graphen, wobei $K' = E \cup F \cup V \cup W$ und $C' = (K' \times K', m_{C'})$[3]:

[3] Im folgenden beziehen sich die Multimengen der direkten Vorgänger und Nachfolger auf die Mengen K' und C'.

- $|W| = \sum_{v \in J_{OR} \cup J_{XOR} \cup J_{AND}} |\{x \in K \mid (x,v) \in C\}|$
- $\forall (x,v) \in C, v \in J_{OR} \cup J_{XOR} \cup J_{AND} \exists w \in W : (w,v) \in C' \wedge (x,w) \in C'$
- $\forall (x,y) \in C, y \in E \cup F \cup S_{OR} \cup S_{XOR} \cup S_{AND} : (x,y) \in C'$
- $\forall (j,k) \in C' : m_{C'}((j,k)) = 1$
- $\forall w \in W : |\bullet w| = |w\bullet| = 1$

Weiterhin sind einige einführende Definitionen erforderlich:

Definition 2.8 (Zustand eines EPK-Schemas) Seien ein Vorverknüpfer-ergänztes EPK-Schema A' und die Menge K' als Menge aller Knoten dieses EPK-Schemas gegeben. $S \in K'_{MS}$ ist ein Zustand des EPK-Schemas A'.

Falls ein Element $e \in K'$ des EPK-Schemas in einem Zustand S enthalten und somit $e \in S$ ist, wird e als aktiviert in S bezeichnet. Für Ereignisse bedeutet dies anschaulich, dass sie eingetreten sind, und für Funktionen, dass sie sich gerade in Ausführung befinden.
Zur Beschreibung der Dynamik einer EPK wird weiterhin zunächst die EPK-Instanz eingeführt, die eine Instanz zu einem gegebenen Schema darstellt und der daher ein konkreter Zustand zugeordnet ist.

Definition 2.9 (EPK-Instanz) Sei S ein Zustand des EPK-Schemas A'. Eine EPK-Instanz I ist ein Tupel $I = (A', S)$.

Zur Definition der Semantik muss exakt festgelegt werden, welche Zustände von einer gegebenen EPK-Instanz $I = (A', S)$ im nächsten Schritt erreicht werden können. Gesucht ist somit eine Transitionsrelation $TR \subseteq K'_{MS} \times K'_{MS}$ für ein EPK-Schema A', die einem Zustand S einen Folgezustand S' zuordnet, so dass $(S, S') \in TR$ gilt. Anschaulich wird dafür im folgenden $S \rightarrow_{TR} S'$ geschrieben. Anhand dieser Transitionsrelation kann der Übergang einer EPK-Instanz $I = (A', S)$ in einen Folgezustand S' ermittelt werden.

Zur Einführung der Transitionsrelation $\rightarrow_{TR}$ wird ein Vorverknüpfer-ergänztes EPK-Schema vorausgesetzt. Sei $S = (K', m_S)$ ein Zustand und $a \in S$. $\{a\}^n$ sei die Multimenge (K', m), die nur das Element a enthält ($b \in \{a\}^n \Rightarrow b = a$) und für die $m(a) = n$ ist. Die Transitionsrelation $\rightarrow_{TR}$ wird folgendermaßen definiert:

$$S \rightarrow_{TR} S' :\Leftrightarrow \begin{cases} \text{1-3. } S' = (S - \{a\}^1) + a\bullet, \text{ falls } a \in (E - E_E) \cup F \cup S_{AND} \\ \text{4. } S' = (S - \{a\}^1) + \{b\}^1,\ b \in a\bullet, \text{ falls } a \in S_{XOR} \\ \text{5. } S' = (S - \{a\}^1) + B,\ B \leq a\bullet \wedge |B| > 0, \text{ falls } a \in S_{OR} \\ \text{6. } S' = (S - \bullet v) + v\bullet, \text{ falls } a \in W,\ v \in a\bullet,\ v \in J_{AND} \text{ und} \\ \quad \forall w \in \bullet v : w \in S \\ \text{7. } S' = (S - \{a\}^1) + v\bullet, \text{ falls } a \in W,\ v \in a\bullet,\ v \in J_{XOR}, \\ \quad |\{w \in \bullet v \mid w \in S\}| = 1 \text{ und} \\ \quad \neg\exists S_1,...,S_n : S_1 = S - \{a\}^1 \wedge S_1 \rightarrow_{TR} ... \rightarrow_{TR} S_n \wedge w \in S_n \wedge w \in \bullet v \\ \text{8. } S' = (S - \bullet v) + v\bullet, \text{ falls } a \in W,\ v \in a\bullet,\ v \in J_{OR} \text{ und} \\ \quad \neg\exists S_1,...,S_n : S_1 = S - \bullet v \wedge S_1 \rightarrow_{TR} ... \rightarrow_{TR} S_n \wedge w \in S_n \wedge w \in \bullet v \end{cases}$$

Der aktuelle Zustand eines EPK-Schemas kann durch „wandernde“ Prozessmappen veranschaulicht werden. Liegt eine Prozessmappe auf einem Kontrollflussobjekt, so ist dieses im Zustand „aktiv“, andernfalls „inaktiv“. Die Transitionsrelation $\rightarrow_{TR}$ beschreibt somit folgende Eigenschaften für die Kontrollflussobjekte:

1. Ereignisse leiten im Zustand „aktiv“ eine Prozessmappe (sofort) an das nachfolgende Kontrollflussobjekt weiter.
2. Funktionen leiten im Zustand „aktiv“ eine Prozessmappe (nach Beendigung des Bearbeitungsvorganges und der Freigabe) an das nachfolgende Kontrollflussobjekt weiter.
3. AND-Split-Operatoren leiten im Zustand „aktiv“ (sofort) genau eine Prozessmappe an jedes nachfolgende Kontrollflussobjekt weiter.
4. XOR-Split-Operatoren leiten im Zustand „aktiv“ (sofort) genau eine Prozessmappe an genau ein nachfolgendes Kontrollflussobjekt weiter.
5. OR-Split-Operatoren leiten im Zustand „aktiv“ (sofort) genau eine Prozessmappe an jedes Kontrollflussobjekt einer Teilmenge der nachfolgenden Kontrollflussobjekte weiter.
6. AND-Join-Operatoren leiten im Zustand „aktiv“ eine Prozessmappe (sofort) an das nachfolgende Kontrollflussobjekt weiter, wenn von allen vorgelagerten Kontrollflussobjekten (Vorverknüpfern) eine Prozessmappe eingetroffen ist.
7. XOR-Join-Operatoren leiten im Zustand „aktiv“ eine Prozessmappe (sofort) an das nachfolgende Kontrollflussobjekt weiter, wenn von genau einem direkt vorgelagerten Kontrollflussobjekt (Vorverknüpfer) eine Prozessmappe eingetroffen ist und keine weiteren Prozessmappen mehr die verbleibenden vorgelagerten Kontrollflussobjekte erreichen können.
8. OR-Join-Operatoren leiten im Zustand „aktiv“ eine Prozessmappe (sofort) an das nachfolgende Kontrollflussobjekt weiter, wenn von jedem Kontrollflussobjekt einer Teilmenge der direkt vorgelagerten Kontrollflussobjekte (Vorverknüpfer) eine Prozessmappe eingetroffen ist und keine weiteren Prozessmappen mehr die verbleibenden vorgelagerten Kontrollflussobjekte erreichen können.

Aufgrund der Semantikdefinition der Verknüpfungsoperatoren kann man leicht eine Kontrollflusslogik beschreiben, die innerhalb einer EPK-Instanz zu einer Unter- bzw. Überbelegung von Join-Operatoren führt. In diesem Fall können Prozessmappen im Verlauf der Prozessausführung im EPK-Schema „verklemmen“ und als Folge kein Endereignis mehr erreichen. Aus pragmatischen Gründen wird im EPK-Kontrollflusskonzept keine generelle Verklemmungsfreiheit gefordert. So kann z.B. zur Abbildung einer maximalen Nebenläufigkeit eine fallbezogenen Blockierung eines Pfades eines EPK-Schemas in der Form einer „Kann-Verklemmung“ beabsichtigt sein. Derartige Überlegungen sind Gegenstand von Validierungs- und Verifikationskonzepten und erfordern Formalismen zur Definition und zum Nachweis allgemeiner Eigenschaften eines EPK-Schemas.

3 Ausblick

In diesem Beitrag wurde für das Kontrollflusskonzept der Ereignisgesteuerten Prozesskette (EPK) eine formale Syntax- und Semantikdefinition entworfen. Diese kann als Grundlage zur Formalisierung des Zeit-, Mengen- und Ressourcenkonzeptes dienen. Alle Konzepte gemeinsam bieten dann eine valide Grundlage für eine integrierte Geschäftsprozessarchitektur zur Konzeption und Implementierung betriebswirtschaftlicher und (informations-)technischer EPK-Anwendungen (Geschäftsprozesssimulation, Prozesskostenrechnung, Workflowmanagement etc.). Ein weiterer wichtiger Nutzen liegt in der Bereitstellung adäquater Modellierungskonventionen und Verifikations- und Transformationsverfahren.

Zur Bündelung der Forschungsaktivitäten haben die Autoren des Beitrages die Gründung des Arbeitskreis „Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten (WI-EPK)“ im Fachbereich Wirtschaftsinformatik der Gesellschaft für Informatik e.V. initiiert. Die Aktivitäten des Arbeitskreises sind unter der URL http://www.epk-community.de dokumentiert.

Literaturverzeichnis

[Aa98] van der Aalst, W.M.P.: Formalization and Verification of Event-driven Process Chains, in: Backhouse, R.C.; Baeten, J.C.M. (Hrsg.): Computing Science Reports 98/01, Eindhoven University of Technology, Eindhoven 1998.

[Aa99] van der Aalst, W.M.P.: Formalization and Verification of Event-driven Process Chains, in: Information and Software Technology 41(1999)10, S. 639-650. URL: http://tmitwww.tm.tue.nl/staff/wvdaalst/Publications/p74.pdf (2002-08-15)

[BRS95] Becker, J.; Rosemann; M.; Schütte, G.: Grundsätze ordnungsmäßiger Modellierung, Wirtschaftsinformatik, 37(1995)5, S. 435-445.

[CS94] Chen, R.; Scheer, A.-W.: Modellierung von Prozeßketten mittels Petri-Netz-Theorie, in: Scheer, A.-W. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 107, Saarbrücken 1994.

[De01] Dehnert, J.: Four Systematic Steps Towards Sound Business Process Models, Proceedings of the 2nd International Colloquium on Petri Net Technologies for Modelling Communication Based Systems, Fraunhofer Gesellschaft ISST, Berlin 2001, pp. 55-64. URL: http://cis.cs.tu-berlin.de/~dehnert/Publikationen/Foursteps.ps (2002-08-15)

[DR01] Dehnert, J.; Rittgen, P.: Relaxed Soundness of Business Processes, Proceedings of the 13th Conference on Advanced Information Systems Engineering (CAISE'01) , Dittrich, K.L. and Geppert, A. and Norrie, M.C. (Eds.), Springer, LNCS 2068, Interlaken, Switzerland, 2001, pp. 157-170. URL: http://cis.cs.tu-berlin.de/~dehnert/Publikationen/CAISE01.ps (2002-08-15)

[He02] Heimig, I.: Grammatikbasierte Beschreibung von Geschäftsprozessen - Methodik für das strukturierte Verarbeiten von Modellen, Deutscher Universitätsverlag, Wiesbaden 2002.

[HSH95] Hoffmann, W.; Scheer, A.-W.; Hoffmann, M.: Überführung strukturierter Modellierungsmethoden in die Object Modelling Technique (OMT), in: Scheer, A.-W. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 114, Saarbrücken 1995.

[Ke99] Keller, G. & Partner: SAP/ R3 prozeßorientiert anwenden. Iteratives Prozeß-Prototyping mit Ereignisgesteuerten Prozeßketten und Knowledge Maps, Addison Wesley Longman, Bonn et al. 1999.

[KNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)", in: Scheer, A.-W. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 89, Saarbrücken 1992. URL: http://www.iwi.uni-sb.de/iwi-hefte/heft089.zip (2002-08-15)

[LSW97a] Langner, P.; Schneider, C.; Wehler, J.: Ereignisgesteuerte Prozeßketten und Petri-Netze, in: Valk, R.; Jantzen, M. (Hrsg.): Bericht Nr. 196, Fachbereich Informatik der Universität Hamburg, Hamburg 1997. URL: http://medoc.informatik.uni-hamburg.de/Dienst/Repository/2.0/Body/ncstrl.uhamburg_cs%2FB-196/postscript (2002-08-15)

[LSW97b] Langner, P.; Schneider, C.; Wehler, J.: Prozeßmodellierung mit ereignisgesteuerten Prozeßketten (EPKs) und Petri-Netzen, in: Wirtschaftsinformatik, 39(1997)5, S. 479-489.

[LSW97c] Langner, P.; Schneider, C.; Wehler, J.: Ereignisgesteuerte Prozeßketten (EPKs) und Petri-Netze, in: Desel, J.; Reichel, H. (Hrsg.): Grundlagen der Parallelität, Workshop der GI-Fachgruppen 0.0.1 und 0.1.7 im Rahmen der INFORMATIK`97, Technische Berichte der TU Dresden, Fakultät Informatik, Dresden 1997.

[LSW97d] Langner, P.; Schneider, C.; Wehler, J.: Relating Event-Driven Process Chains to Boolean Nets, Ludwig-Maximilians-Universität München, Institut für Informatik, Technischer Bericht 9707, München 1997. URL: www.pst.informatik.uni-muenchen.de/personen/wehler/tecrep3.ps (2002-08-15)

[LSW98] Langner, P.; Schneider, C.; Wehler, J.: Petri Net Based Certification of Event driven Process Chains, in: Desel, J.; Silva, M. (Hrsg.): Application and Theory of Petri Nets 1998, Lecture notes in Computer Science Vol. 1420, (Springer) Berlin et al. 1998, S. 286-305.

[MR00] Moldt, D., Rodenhagen, J.: Ereignisgesteuerte Prozeßketten und Petrinetze zur Modellierung von Workflows, in: Giese, H.; Philippi, S. (Hrsg.): Visuelle Verhaltensmodellierung verteilter und nebenläufiger Software-Systeme, Proceedings (Münster, 13.-14. November 2000), 8. Workshop des Arbeitskreises "Grundlagen objektorientierter Modellierung" (GROOM) der GI-Fachgruppe 2.1.9 ("Objektorientierte Softwareentwicklung"), Bericht Nr. 24/00-I, Münster 2000, S. 57-63. URL: http://wwwmath.uni-muenster.de/cs/u/versys/workshops/VVVNS2000/papers/Moldt.pdf (2002-08-15)

[Mo96] Moll, M.: Formalisierung und Konsistenzprüfung des R/3-Referenzmodells, Diplomarbeit im Studiengang Informatik an der TU Clausthal (Prof. Möller), Clausthal 1996.

[Ri99a] Rittgen, P.: Vom Prozessmodell zum Elektronischen Geschäftsprozess, in: Frank, U.; Hampe, F.: Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 17, Koblenz-Landau 1999. URL: http://www.uni-koblenz.de/~rittgen/Nr17.pdf (2002-08-15)

[Ri99b] Rittgen, P.: Modified EPCs and Their Formal Semantics, in: Frank, U.; Hampe, F.: Arbeitsberichte des Instituts für Wirtschaftsinformatik, Nr. 19, Koblenz-Landau 1999. URL: http://www.uni-koblenz.de/~rittgen/Nr19.pdf (2002-08-15)

[Ri99c] Rittgen, P.: Objektorientierte Analyse mit EMK, in: Sinz, E.J. (Hrsg.): Modellierung betrieblicher Informationssysteme, Proceedings der MobIS-Fachtagung 1999 (Bamberg, 14.-15.10.1999), Rundbrief der GI-Fachgruppe 5.10 6(1999)1, S. 8-23. URL: http://www.uni-koblenz.de/~rittgen/mobis99.pdf (2002-08-15)

[Ri99d] Rittgen, P.: From Process Model to Electronic Business Process, in: European Conference on Information Systems ECIS 1999, June 23 - 25, Copenhagen Business School, Copenhagen, Denmark, proceedings, volume II, 1999, pp. 616 ff. URL: http://www.uni-koblenz.de/~rittgen/ecis99.pdf (2002-08-15)

[Ri00a] Rittgen, P.: Paving the Road to Business Process Automation, European Conference on Information Systems (ECIS) 2000, Vienna, Austria, July 3 - 5, 2000, pp. 313-319. URL: http://www.uni-koblenz.de/~rittgen/ecis00.pdf (2002-08-15)

[Ri00b] Rittgen, P.: EMC - A Modeling Method for Developing Web-based Applications, International Conference of the International Resources Management Association (IRMA) 2000, Anchorage, Alaska, USA, May 21 - 24, 2000, pp. 135-140. http://www.uni-koblenz.de/~rittgen/irma00.pdf (2002-08-15)

[Ri00c] Rittgen, P.: Quo vadis EPK in ARIS? Ansätze zu syntaktischen Erweiterungen und einer formalen Semantik, in: WIRTSCHAFTSINFORMATIK 42(2000)1, S. 27-35. URL: http://www.uni-koblenz.de/~rittgen/ZWI00.pdf (2002-08-15)

[Ro97] Rodenhagen, J.: Darstellung ereignisgesteuerter Prozeßketten (EPK) mit Hilfe von Petrinetzen, Diplomarbeit Universität Hamburg Fachbereich Informatik (Prof. Valk), Hamburg 1997.

[Ro96] Rosemann, M.: Komplexitätsmanagement in Prozeßmodellen - Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung, Dissertation Universität Münster (Prof. Becker), in: Scheer, A.-W. (Hrsg.): Schriften zur EDV-orientierten Betriebswirtschaftslehre, Gabler, Wiesbaden 1996.

[Ru95] Rump, F.J.: Ereignisgesteuerte Prozeßketten zur formal fundierten Geschäftsprozeß-modellierung, in: Informationssystem-Achitekturen, Rundbrief des GI-Fachausschusses 5.2, 2(1995)2, S. 94-96.

[Ru97a] Rump. F.J.: Erreichbarkeitsgraphbasierte Analyse ereignisgesteuerter Prozeßketten, Technischer Bericht Fachbereich Informatik Universität Oldenburg, Oldenburg 1997.

[Ru97b] Rump, F.J.: Analysis of business process descriptions, in: Pokorny, J. (Hrsg.): Proceedings of the 17th Annual Database Conference DATASEM´97, Brno, Czech republic, October 1997.

[Ru97c] Rump, F.J.: Analyse ereignisgesteuerter Prozeßketten (EPK), in: Desel, J.; Reichel, H. (Hrsg.): Grundlagen der Parallelität, Workshop der GI-Fachgruppen 0.0.1 und 0.1.7 im Rahmen der INFORMATIK´97, Technische Berichte der TU Dresden, Fakultät Informatik, Dresden 1997

[Ru99] Rump, F.: Geschäftsprozeßmanagement auf der Basis ereignisgesteuerter Prozeßketten - Formalisierung, Analyse und Ausführung von EPKs, Teubner, Stuttgart et al.1999.

[Sc99] Scheer, A.-W.: ARIS - Vom Geschäftsprozeß zum Anwendungssystem, 4. Aufl., (Springer) Berlin et al. 1998.

[Sc01] Scheer, A.-W.: ARIS - Modellierungsmethoden, Metamodelle, Anwendungen, 4. Aufl., (Springer) Berlin et al. 2001.

[Ut97] v. Uthmann, Chr.: Nutzenpotentiale der Petrinetztheorie für die Erweiterung der Anwendbarkeit Ereignisgesteuerter Prozeßketten, Workshop der Arbeitsgruppe "Formalisierung und Analyse Ereignisgesteuerter Prozeßketten (EPK)", Universität Oldenburg, Oldenburg 1997. URL: http://www.wi.uni-muenster.de/is/mitarbeiter/ischut/epkpn.pdf (2002-08-15)

[Ut98] v. Uthmann, Chr.: Machen Ereignisgesteuerte Prozeßketten (EPK) Petrinetze für die Geschäftsprozeßmodellierung obsolet?, in: EMISA FORUM (Hrsg.): Mitteilungen der GI-Fachgruppe "Entwicklungsmethoden für Informationssysteme und deren Anwendung", 1(1998), S. 100-107.

[Vo97] Volkmer, M.: Entwicklung objektorientierter Analysemodelle für Informationssysteme auf Grundlage von Prozessmodellen, Shaker, Aachen 1997.

[We97] Weikum, G.; Wodtke, D.; Kotz Dittrich, A.; Muth, P.; Weißenfels, J.: Spezifikation, Verifikation und verteilte Ausführung von Workflows in MENTOR, in: Informatik Forschung und Entwicklung, 1(1997)12, Springer, Berlin et al. 1997. URL: http://www-dbs.cs.uni-sb.de/papers/ife97.ps (2002-08-15)

[Wo97] Wodtke, D.: Modellbildung und Architektur von verteilten Workflow-Management-Systemen, Infix, Sankt Augustin 1997.

[ZR96] Zukunft, O.; Rump, F.: From Business Modelling to Workflow Management: An Integrated Approach, in: Scholz-Reiter, B.; Stickel, E. (Hrsg.): Business Process Modelling, Springer, Berlin et al. 1996, S. 3-22.