

# Event-Driven-Process-Chain-Markup-Language (EPML): Anforderungen, Konzeption und Anwendung eines XML-Schemas für Ereignisgesteuerte Prozessketten (EPK)

Jan Mendling  
Markus Nüttgens (Betreuer)

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

**Abstract:** Mit dem hier vorgeschlagenen Ansatz lässt sich die Syntax der EPK als EPML-Format formal beschreiben. Eine wichtige Rolle spielt dabei die Unterscheidung von impliziten und expliziten Objekttypen. Bei der Konvertierung zu EPML ist es möglich mit dem skizzierten Algorithmus für ein EPK-Modell als Menge an expliziten Objekttyp-Instanzen deren impliziten Objekttyp-Instanzen zu bestimmen. Somit bietet EPML neben der Nutzung als intermediäres Austauschformat auch die Gewährleistung der syntaktischen Richtigkeit.

## 1 Notwendigkeit eines XML-Schemas für EPKs

Bei der Modellierung von Geschäftsprozessen mit Ereignisgesteuerten Prozessketten (EPK) stellt sich die Frage, auf welches Modellierungswerkzeug zurückgegriffen werden soll. Kriterien für die Auswahl finden sich unter anderem bei [Nu02]. Durch die Heterogenität der Werkzeuge sowohl in ihrem Persistenzformat als auch in ihrer Ausdrucksstärke ergeben sich jedoch Probleme beim Austausch von Modellen zwischen verschiedenen Tools, wie es etwa unter Projektpartnern oder bei der Konsolidierung nach einem Unternehmenszusammenschluss nötig wird. [WHB02] beschreiben Lösungsstrategien für dieses Problem. Bei einer Vielzahl verschiedener Formate erscheint ein intermediärer Standard in XML zum Modellaustausch als sinnvoll [MN02], der in dieser Diplomarbeit mit XML Schema definiert werden soll. Als Konvertersprache kann dann XSLT genutzt werden.

Ziel dieser Diplomarbeit ist es, auf [NR02] aufbauend die EPK auf ein syntaktisches Fundament zu stellen. Die Anforderungen für die Definition der Syntax werden anschließend erarbeitet und in XML Schema als „Event-Driven-Process-Chain-Markup-Language“ (EPML) implementiert. Daraus ergibt sich die Möglichkeit, die EPK einer syntaktischen Verifikation zu unterziehen, was über den Ansatz von [GK02] hinausgeht. Des Weiteren wird mit EPML ein intermediärer Standard für einen methodisch orientierten Austausch von EPK-Modellen bereitgestellt.

## 2 Syntaktische Anforderungen an eine EPML

Seit der erstmaligen Beschreibung Ereignisgesteuerter Prozessketten (EPK) [KNS92] ist das Konzept kontinuierlich weiter formalisiert worden. An dieser Stelle seien die Begriffe explizite und implizite Objekttypen vorgestellt. Explizite Objekttypen sind die aus [KNS92] und [NR02] bekannten Standard-Objekttypen Ereignis, Funktion, AND-Konnektor, OR-Konnektor, XOR-Konnektor, Prozesswegweiser und Kontrollflusskante. Bei diesen expliziten Objekttypen handelt es sich zum Teil um Generalisierungen von an dieser Stelle so bezeichneten impliziten Objekttypen. Den Ausführungen von [KNS92] folgend werden für Konnektoren 10 erlaubte Kombinationen unterschieden. Für den AND-Konnektor etwa sind die 4 Kombinationen EreignisFunktionJoin(AND), FunktionEreignis-

Join(AND), EreignisFunktionSplit(AND) als auch der FunktionEreignisSplit(AND). Sie seien als implizite Objekttypen bezeichnet. Analog sind für die anderen beiden Konnektoren jeweils 3 Kombinationen erlaubt. Gleichmaßen sind für Ereignisse die impliziten Typen Startereignis und Endereignis vorhanden, und für den Prozesswegweiser die impliziten Typen Prozesswegweiser(Start) und Prozesswegweiser(Ende). Da die hierarchisierte Funktion keinen Einfluss auf die syntaktische Symbolfolge hat wird hier von einem einzigen impliziten Objekttyp für die Funktion ausgegangen, welcher dem expliziten Objekttyp 1 : 1 entspricht.

Die obigen Konstrukte, die hier als implizite Objekttypen bezeichnet werden, sind aus der Literatur bekannt. Für die Formalisierung der Syntax fehlt ein solches Konzept für den expliziten Objekttyp Kante. Über die impliziten Objekttypen von 10 Konnektoren, 3 Ereignissen, 2 Prozesswegweisern und einer Funktion ergibt sich eine nicht-symmetrische Matrix von 16 x 16 Von-Nach-Verknüpfungen, von denen 88 erlaubt sind. Hier zeigt sich, dass die oft beschriebene leicht verständliche Ausdrucksfähigkeit sich durch eine implizite Komplexität erklärt.

Für die syntaktische Prüfung eine EPK lassen sich die impliziten Kantentypen auf vier verschiedene reduzieren, nämlich EreignisFunktionKante (EFK), FunktionEreignisKante (FEK), ProzesswegweiserEreignisKante (PwEK) und EreignisProzesswegweiserKante (EPwK). Sie dürfen folgende Objekttypen-Gruppen verbinden (siehe Tab.1).

Kante	Von	Nach
EFK	Ereignis Ereignis-Funktion-Join Ereignis-Funktion-Split	Funktion Ereignis-Funktion-Join Ereignis-Funktion-Split
FEK	Funktion Funktion-Ereignis-Join Funktion-Ereignis-Split	Ereignis Funktion-Ereignis-Join Funktion-Ereignis-Split
PwEK	Prozesswegweiser	Ereignis
EPwK	Ereignis	Prozesswegweiser

Tab. 1: Kantentypen und von ihnen verknüpfte Objekte

### 3 Implementierung in XML Schema und Anwendung

Aus den obigen Anforderungen ergeben sich die Definitionsobjekte von EPML in XML Schema. EPML lässt sich als intermediäres Format zum Austausch von EPK-Modellen zwischen heterogenen Modellierungstools einsetzen. Dafür sind die XML-Exportdateien des Modellierungstools via XSLT in EPML um zu wandeln, und davon ausgehend in ein anderes proprietäres XML-Format.

In Tools werden lediglich die expliziten Objekttypen zur Modellierung bereitgestellt. Daher müssen bei der Konvertierung nach EPML die impliziten Objekttypen algorithmisch bestimmt werden. Dabei wird ein Algorithmus verwendet, der aus dem Modell aus expliziten Objekttyp-Instanzen das korrespondierende Modell aus impliziten Objekttyp-Instanzen generiert. Die Grundidee sei folgend kurz dargestellt.

Nicht-trivial ist die Bestimmung des Objekttyps einer Konnektor-Instanz. Dabei kann jedoch auf die Eigenschaft eines Konnektors zurückgegriffen werden, dass er mindestens ein Ereignis und eventuell auch eine Funktion in der Menge seiner transitiven Vorgänger hat. Vom Konnektor ausgehend handelt man sich zurück, bis man auf einen oder mehrere Nicht-Konnektoren trifft. Wenn deren Typ unterschiedlich ist, breche ab mit Meldung eines Syntax-Fehlers. Ist deren Typ gleich, so bestimmt sich daraus und aus der Prüfung, ob mehrere Kanten ein- oder ausgehen, der implizite Typ des Konnektors. Ergibt sich beim Heraufhängeln ein Zyklus, so ist dies ein Konnektor-Zyklus: breche ab mit Syntax-Fehler.

#### 4 Zusammenfassung

Mit dem vorgeschlagenen Ansatz lässt sich die Syntax der EPK als EPML-Format formal beschreiben. Eine wichtige Rolle spielt dabei die Unterscheidung von impliziten und expliziten Objekttypen. Bei der Konvertierung zu EPML ist es möglich mit dem skizzierten Algorithmus für die Menge an expliziten Objekttyp-Instanzen deren impliziten Objekttyp-Instanzen zu bestimmen. Somit bietet EPML neben der Nutzung als intermediäres Austauschformat auch die Gewährleistung der syntaktischen Richtigkeit.

#### Literaturverzeichnis

- [GK02] Geissler, M.; Krüger, A.: Eine XML-Notation für Ereignisgesteuerte Prozessketten (EPK), in: Nüttgens, M.; Rump, F.J. (Hrsg.): Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten - EPK 2002, Proceedings des GI-Workshops und Arbeitskreistreffens (Trier, November 2002), S. 81-86.
- [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. Abruf von <http://www.iwi.uni-sb.de/iwi-hefte/heft089.zip> am 20.10.2002.
- [MN02] Mendling, J.; Nüttgens, N.: Event-Driven-Process-Chain-Markup-Language (EPML): Anforderungen zur Definition eines XML-Schemas für Ereignisgesteuerte Prozessketten (EPK), in: Nüttgens, M.; Rump, F.J. (Hrsg.): Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten - EPK 2002, Proceedings des GI-Workshops und Arbeitskreistreffens (Trier, November 2002), S. 87-93.
- [NR02] Nüttgens, M.; Rump, F.J.: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK), in: Desel, J.; Weske, M. (Hrsg.): Promise 2002 - Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen, Proceedings des GI-Workshops und Fachgruppentreffens (Potsdam, Oktober 2002), LNI Vol. P-21, Bonn 2002, S. 64-77.
- [Nu02] Nüttgens, M.: Rahmenkonzept zur Evaluierung von Modellierungswerkzeugen zum Geschäftsprozessmanagement. In (Gesellschaft für Informatik (GI) e.V.): Informationssystem-Architekturen, Wirtschaftsinformatik Rundbrief der GI Fachgruppe WI-MobIS, 9, 2002.
- [WHB02] Wüstner, E.; Hotzel, T.; Buxmann, P.: Converting Business Documents: A Classification of Problems and Solutions using XML/XSLT. In: Proceedings of the 4th International Workshop on Advanced Issues of E-Commerce and Web-based Systems (WECWIS 2002).