

**Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering**

**Konzept zur Verknüpfung von
Use Cases mit ereignisgesteuerten Prozessketten**

Masterarbeit

im Studiengang Informatik

von

Dimitri Diegel

**Erstprüfer: Prof. Dr. Kurt Schneider
Zweitprüfer: Prof. Dr. Michael H. Breitner
Betreuer: Dipl. Wi.-Inf. Daniel Lübke**

Hannover, 20 Oktober 2006

Inhaltsverzeichnis

I. Einführung	4
I.1 Motivation	4
I.2 Aufgabenstellung	4
I.3 Aufbau	5
1. Grundbegriffe	6
1.1 Use Case	6
1.2 Ereignisgesteuerte Prozesskette	10
1.3 Zusammenfassung	13
2. Definition der Abbildungen	14
2.1 Abbildung eines Use Case auf eine EPK	14
2.2 Abbildung einer EPK auf ein Use Case	20
2.3 Zusammenfassung	33
3. Formate	35
3.1 XML und XML Schema	35
3.2 Definition eines Formats für Use Cases	36
3.3 EPML das Format für EPKs	40
3.4 Zusammenfassung	41
4. Implementierung der Abbildungen	42
4.1 XSLT	42
4.2 Transformation der Use Cases	44
4.3 Transformation der EPKs	56
4.4 User Interface	64
4.5 Zusammenfassung	66
5. Anwendungsszenario	67
5.1 Anlegen und Transformieren der Use Cases	67
5.2 Anlegen und Transformieren der EPKs	73
6. Schluss	78
6.1 Verwandte Arbeiten	78
6.2 Zusammenfassung	79
6.3 Ausblick	80
7. Literaturverzeichnis	81

I. Einführung

I.1 Motivation

Um konkurrenzfähig zu bleiben und die Kosten zu senken sind die Unternehmen ständig bemüht ihre geschäftlichen Abläufe zu optimieren. Damit die Geschäftsprozesse eines Unternehmens analysiert, bewertet und schließlich optimiert werden können, müssen sie erst einmal dokumentiert werden. Für die Modellierung und damit auch die Dokumentation von Geschäftsprozessen hat sich heutzutage das auf ereignisgesteuerten Prozessketten (EPK) basierte ARIS¹- Konzept als internationaler Standard durchgesetzt [AWScheer2001].

Die Optimierung von Geschäftsprozessen erfolgt nicht selten dadurch, dass Teile dieser Prozesse durch Softwaresysteme automatisiert werden. Die Erhebung und Fixierung von funktionalen Anforderungen ist der erste Schritt im Entwicklungszyklus jedes Softwaresystems. Requirements Engineering hat sich inzwischen zu einer eigenen Disziplin innerhalb des Software Engineering entwickelt, sie befasst sich mit den Techniken und Methoden der Aufnahme, Analyse, Dokumentation und Management von Anforderungen. Die Use Cases (Anwendungsfälle) ist eine weit² verbreitete Technik auf dem Gebiet des Requirements Engineering zur Dokumentation von funktionalen Anforderungen.

Die mit den EPKs modellierten Geschäftsprozesse sind also oft Teil eines Softwareentwicklungsprojektes. Die betriebswirtschaftlichen Abläufe und Zusammenhänge stehen jedoch im Vordergrund der Geschäftsprozessmodellierung, unabhängig davon wie sie mit Soft- oder Hardware unterstützt werden. Die mit den EPKs definierten Anforderungen sind deshalb für die Entwickler von Softwaresystemen nicht detailliert genug und müssen oft mit den Methoden des Requirements Engineering präzisiert werden [GeUML2003]. Die Beschleunigung dieses Prozesses durch computergestützte Tools sowie die explizite Zuordnung zwischen EPKs und entsprechenden Use Cases anhand von offenen Standards wäre daher äußerst wünschenswert.

I.2 Aufgabenstellung

Das Ziel dieser Arbeit ist die Entwicklung und Implementierung eines Konzepts mit dem eine Zuordnung zwischen den Anforderungen in Form von Use Cases und den entsprechenden Prozessen in Form von EPKs vorgenommen werden kann. Dieses Konzept soll es also ermöglichen sowohl aus Use Cases die entsprechende EPK- Darstellung zu erzeugen als auch umgekehrt aus den EPKs die entsprechenden Use Cases zu generieren. Um die Integration in möglichst viele computergestützte Werkzeuge zu erleichtern oder überhaupt zu ermöglichen, soll die Implementierung auf einem offenem, allgemein anerkanntem Datenaustauschformat basieren, also auf XML³. Die Beschreibung der notwendigen Transformation in XML setzt voraus, dass sowohl die EPKs als auch die Use Cases in einem XML- Format vorliegen. Ein XML- Format für die EPKs wurde bereits in [EPK2002 Seite 81] vorgeschlagen, ein XML-

¹ Architektur Integrierter Systeme

² Die Use Cases sind fester Bestandteil vieler heutzutage gebräuchlichen Softwareentwicklungsmodelle:

Wasserfallmodell, Rational Unified Process, Extreme Programming

³ Extensible Markup Language

Format für die Use Case liegt noch nicht vor und soll deshalb in dieser Arbeit ebenfalls entwickelt werden.

Die Use Cases und die EPKs haben zwar Gemeinsamkeiten, sind aber dennoch nicht vollkommen äquivalent, wie sich im Laufe dieser Arbeit herausstellen wird. Eine vollautomatische Übersetzung der EPKs in entsprechende Use Cases und umgekehrt wird also ohne gewisse Benutzereingabe kaum möglich sein. Sowohl die Ausarbeitung welche Informationen durch den Menschen eingegeben werden müssen, als auch die Entwicklung eines Prototypen mit dem die Eingabe dieser Informationen vorgenommen werden kann ist ein weiteres Ziel der vorliegende Arbeit.

I.3 Aufbau

Im ersten Kapitel wird das Konzept der Use Cases und der EPKs vorgestellt und näher erläutert. Die Use Cases und die EPKs bilden die Grundlage für diese Arbeit.

Im zweiten Kapitel werden die Ausdrucksmöglichkeiten der Use Cases und EPKs miteinander verglichen und auf die Unterschiede und Gemeinsamkeiten dieser Konzepte eingegangen. Ausgehend von diesem Vergleich werden Abbildungsvorschriften vorgeschlagen, die eine Zuordnung zwischen den Elementen dieser beiden Modelle ermöglichen.

Im dritten Kapitel werden die XML Formate für Use Cases und EPKs vorgestellt, EPML¹ wurde von [MendlingNüttgens] veröffentlicht, und stellt ein Konzept zur Abspeicherung der EPKs in Form einer XML Datei dar, da für Use Cases noch kein XML Format vorliegt, ist die Ausarbeitung eines Use Case Formats ein weiterer Meilenstein dieses Kapitels.

XSLT² ist eine Scriptsprache, die entwickelt wurde um Transformationen von XML Dokumenten zu beschreiben, da sowohl für die Use Cases als auch für die EPKs ein XML-Schema (das im 3 Kapitel vorgestellt worden ist) vorliegt, wird XSLT verwendet um die in Kapitel 2 definierten Transformationsvorschriften umzusetzen. Kapitel 4 enthält eine kurze Einführung in XSLT sowie die Dokumentierung der Implementierung dieser Vorschriften.

Die XML Beschreibungen von Use Cases und besonders von EPKs können von den Menschen nur mit viel Mühe gelesen und verstanden werden. Im Rahmen dieser Arbeit wird deshalb ein Prototyp entwickelt mit dem EPKs und Use Cases erstellt und in einander überführt werden können, die Benutzung dieses Programms sowie die Vorzüge des in dieser Arbeit entwickelten Konzepts werden anhand eines Anwendungsszenario im fünften Kapitel demonstriert.

Das sechste Kapitel bietet sowohl eine Zusammenfassung der Ergebnisse, die in dieser Arbeit erzielt wurden als auch einen Ausblick auf weitere Arbeiten, die in dieser Richtung bereits betrieben wurden und in Zukunft noch werden könnten.

¹ Eventdriven Processchain Modelling Language

² eXtensible Stylesheet Transformation Language

6 Schluss

6.1 Verwandte Arbeiten

Zur Erhöhung der Produktivität und Senkung der Kosten ist es heutzutage üblich die Geschäftsprozesse in betriebliche Informationssysteme umzusetzen, dabei entstehen häufig Probleme, weil die Methoden, Technologien und Werkzeuge, die von den Softwareentwickler verwendet werden sich unabhängig von denen der Geschäftsprozessmodellierung entwickelt haben und deshalb deutliche Unterschiede aufweisen. Um dieses Problem zu entschärfen werden in zahlreichen Veröffentlichungen Strategien zur parallelen Benutzung oder der Übersetzung der Softwareentwicklungs- und Geschäftsprozessmodellierungsmethoden propagiert:

- So wird in [GeUML2003] ein kompletter Leitfaden zur Modellierung der Geschäftsprozesse mit den Methodiken der UML vorgestellt. Das Werk beschreibt Praktiken über die Erstellung, Vervollständigung und Verfeinerung der Geschäftsprozess- Use Cases bis hin zur Modellierung der Unternehmensstruktur und Abläufe mit den UML Klassen-, Aktivitäts- und Zustandsdiagrammen.
- In [AC2003] wird ein Kapitel der Modellierung von Geschäftsprozessen mit den Use Cases gewidmet.
- In [LoosAllweyer1998], [NFZ1998] und [Dandl1999] werden EPK Diagramme mit Klassen, Sequenz-, Aktivitäts- und Use Case Diagrammen (aber nicht Use Cases selbst) verglichen und einige Ideen skizziert, wie EPK Diagramme und die oben genannten UML Diagramme in einander transformiert werden können.
- In [SNZ1997] wird eine objektorientierte Erweiterung für EPKs vorgestellt, die es dem Modellierer erlaubt Klassen und Methoden, wie sie in einem UML Klassendiagramm verwendet werden, darzustellen.

Alle diese Arbeiten bieten Methoden und Praktiken an, die nicht formal definiert sind, sondern eher an die menschliche Intuition und das umfassende Verständnis der entsprechenden Sachverhalte appellieren. Einen anderen Ansatz verfolgen folgende Arbeiten, zu denen auch diese gehört. Hier werden ebenfalls Methoden vorgeschlagen, die etwas mit der Transformation von Geschäftsprozessen in Form von EPKs zu tun haben, allerdings sind diese Methoden so konzipiert, dass sie nicht manuell durch der Menschen sondern automatisch oder zumindest halbautomatisch durch den Computer ausgeführt werden sollen.

- In [LLSG2006] werden EPK Erweiterungen vorgeschlagen, die es ermöglichen aus Ereignisgesteuerten Prozessketten Webservices und Userinterfaces zu generieren.
- Zum Zwecke der Verifikation und Simulation wird in [VDA] ein Verfahren zur Transformation der EPKs in Petri Netze präsentiert.
- Im [MZ] wird ein Verfahren diskutiert, wie aus BPEL Beschreibungen EPK Diagramme erzeugt werden können.

- Eine Transformationsmethode von EPK Modellen in das POP* Format ist in der Arbeit von [KK] zu finden.

6.2 Zusammenfassung

Um Use Cases und EPK Diagramme in einander konvertieren zu können ist zunächst eine genaue Definition der Syntax dieser beiden Sprachen erforderlich. Im Falle der EPKs wurde die Syntax in der Arbeit von [MendingNüttgens] in Form eines XML Schema definiert, weil für die Use Cases keine Syntax Definition vorlag, wurde in dieser Arbeit ein XML Schema entwickelt das die Syntax der Use Cases nach dem Buch von [AC2003] beschreibt.

Wegen der unkomplizierten Handhabung und des hohen Verbreitungsgrades wurde die XSLT-Technologie gewählt um die notwendigen Transformationen zur Konvertierung zwischen Use Cases und EPKs zu beschreiben. Es hat sich dabei herausgestellt, dass die einfache Form der Use Cases, die auch in dem oben erwähnten Buch von [AC2003] propagiert wird, problemlos und ohne jegliche Benutzereingaben sich auf ein EPK Diagramm abbilden lässt. Bei der umgekehrten Abbildung, also der Abbildung eines EPK Diagramms auf ein Use Case hat sich gezeigt, dass bestimmte Benutzereingaben unerlässlich sind, das heißt also, dass die Use Cases gewisse Informationen (vergleiche Kapitel 2.2 und 5.2) enthalten, die durch die EPK Diagramme nicht erfasst werden, woraus die Nichtäquivalenz zwischen Use Cases und EPKs folgt.

Weiterhin wurde festgestellt, dass die EPK Diagramme, die durch Transformation der Use Cases entstehen, alle ein bestimmtes Muster (Kapitel 2.2) aufweisen, die Abbildung der Use Case Menge auf die Menge der EPKs ist deshalb nicht surjektiv (es gibt solche EPKs zu denen kein passender Use Case als Urbild existiert).

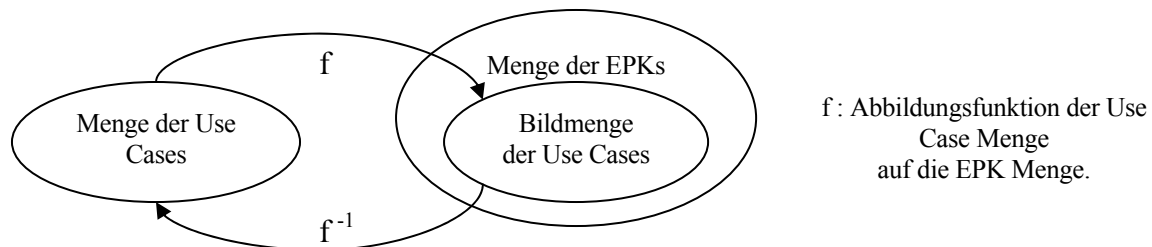


Abbildung 6-1

Um die Abbildungsvorschriften für die EPKs zu definieren, reichte es also nicht aus einfach die Umkehrabbildung der Use Cases zu nehmen, da mit dieser Umkehrabbildung nicht alle EPKs erwischt worden wären, wie in der Abbildung 6-1 zu sehen ist. Um dieses Problem zu beheben wurde eine weitere Abbildung angegeben, mit der die EPKs in die Bildmenge der Use Cases abgebildet werden können. Damit wurde schließlich erreicht, dass jede EPK auf ein Use Case abgebildet werden konnte (in dem sie zunächst in die Bildmenge und dann mit der Umkehrabbildung in die Menge der Use Cases abgebildet wurde, siehe auch Abbildung 6-2).

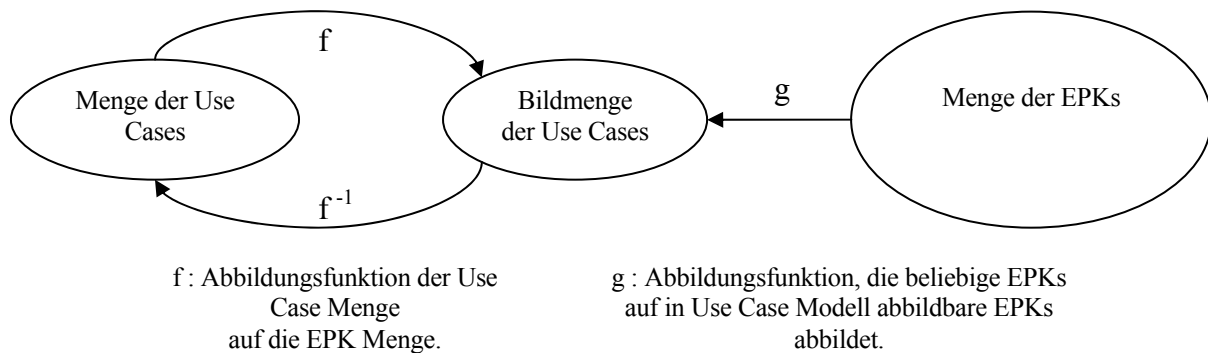


Abbildung 6-2

Da sowohl die Use Cases als auch die EPKs im XML Format vorlagen, der für den Menschen recht unübersichtlich ist, wurde im Rahmen dieser Arbeit eine Benutzeroberfläche entwickelt, mit der es möglich ist Use Cases und EPKs graphisch (nicht auf XML Ebene) zu erstellen und in einander zu überführen.

6.3 Ausblick

Das Konzept der Konvertierung und der Verknüpfung zwischen Use Cases und EPKs kann in vielen Situationen hilfreich sein. Die Konvertierung der EPKs in Use Cases kann z.B. für die Dokumentation der EPKs verwendet werden. Solche Dokumentationsform wird sowohl von den Softwareentwicklern als auch von Wirtschaftswissenschaftlern verstanden und ist vor allem dann hilfreich wenn die betriebswirtschaftlichen Prozesse durch Software automatisiert werden und diese beiden Gruppen aufeinander treffen. Aber auch dann wenn die Use Cases in Projekten verwendet werden, die mit Geschäftsprozessen nichts zu tun haben, hilft die Konvertierung in die EPK Diagramme den Überblick über die zusammenhängenden Use Cases zu verschaffen und ist somit bei der Validierung der Use Cases hilfreich.

In der vorliegenden Arbeit wurde deshalb ein Konzept vorgeschlagen und implementiert, mit dem vollautomatisch Use Cases in EPKs konvertiert werden können und eine halbautomatische Konvertierung der EPKs in Use Cases möglich ist. Diese Konvertierung ist allerdings nur auf solche EPKs beschränkt, die aus den EPK- Basiselementen (Ereignis, Funktion, Prozesswegweiser und AND-, OR-, XOR- Konnektoren) bestehen.

Das Konzept der EPKs wurde inzwischen um zahlreiche Erweiterungen (die in dieser Arbeit nicht berücksichtigt wurden) ergänzt, zu diesen Erweiterungen gehören z.B. die Aris-Sprachkonstrukte [ScheerThomas2005], die Finanzmittel, menschliche Arbeitsleistung, maschinelle Ressourcen, Computerhardware, Anwendungssoftware etc. repräsentieren. Mit der objektorientierten Erweiterung [SNZ1997] lassen sich auch Klassen und ihre Methode in einer EPK darstellen. Für die Zukunft wäre es also auch interessant zu untersuchen inwieweit diese Erweiterungen sich auf UML Konstrukte (nicht nur auf Use Cases) abbilden lassen. Insbesondere die objektorientierte Erweiterung mit ihren Klassen und Methoden, die auch in UML- Klassendiagrammen modelliert werden, verspricht gute Integrationsmöglichkeiten.