

Geschäftsprozessmodellierung und -optimierung mit Methoden der EPK, Petrinetze und der UML.

Erstellt im Rahmen eines Praktikums bei der T-Systems International GmbH.

Manuskript: Sven Kölsch (SK) <mail@sven-koelsch.de>,

21.04.-16.06.2004

Version 0.08 \LaTeX

21. Juni 2004

Inhaltsverzeichnis

1	Definitionen	3
1.1	Definition Geschäftsprozess	3
1.2	Definition Prozessorientierung	3
1.3	Definition Business Process Reengineering	3
1.4	Ziele der GPM	4
1.5	Vorgehen:	4
1.5.1	Probleme:	4
1.6	Methodenübersicht zur Beschreibung von Prozessen	5
2	Modellierungsziel: Prozessoptimierung	6
2.1	Phase 1: Prozessanalyse	6
2.2	Phase 2: Optimierungsziel festlegen	7
2.3	Phase 3: Prozessmodellierung	7
2.4	Phase 4: Optimierung	7
3	Komponenten zur Prozessbeschreibung	8
3.1	Ereignisgesteuerte Prozessketten (EPK)	8
3.1.1	Funktionen	8
3.1.2	Ereignisse	8
3.1.3	Organisationseinheiten	9
3.1.4	Informationsobjekte	9
3.1.5	Verknüpfung durch Operatoren und Kontrollfluss	9
3.1.6	Software	9
3.2	Petrinetze	10
3.2.1	Stellen	10
3.2.2	Transitionen	11
3.2.3	Kanten	11
3.2.4	Token	11
3.2.5	Software	11
3.3	UML Aktivitätsdiagramme	12
3.3.1	Aktivität	12
3.3.2	Übergang (engl. hier: Transition)	13
3.3.3	Verzweigung (Branch)	13

<i>INHALTSVERZEICHNIS</i>	2
3.3.4 Zusammenführung (Branch, Boolean: ODER)	13
3.3.5 Teilung (Fork)	14
3.3.6 Synchronisation (Join, Boolean: UND)	14
3.3.7 Objektknoten (Object Flow State)	14
3.3.8 Software	14
A Anhang	18
A.1 Vergleich eEPK, Petrinetz und UML-Aktivitätsdiagramm	19

Kapitel 1

Definitionen

1.1 Definition Geschäftsprozess

- In der Literatur nicht einheitlich [WIK, Staud2001]:
 1. Anlehnung an Business Process Reengineering (s.u.): Geschäftsprozesse sind Kernprozesse, die das Leistungsspektrum eines Unternehmens darstellen \Rightarrow Ergebnis: Wert für den Kunden [WIK, HaCha1995]
 2. Allgemeineres Verständnis: Geschäftsprozesse sind betriebliche Prozesse, die zur Erstellung der Unternehmensleistung beitragen (u.a. auch Produktentwicklung u. Marketing)[WIK, KeTeu1997]
 3. Systemtheoretische Sichtweise: Geschäftsprozesse sind Folgen bestimmter diskreter Zustandsänderungen des betrachteten Systems Unternehmen [WIK] oder auch als ein Subsystem der Ablauforganisation, dessen Elemente Aufgaben, Aufgabenträger und Sachmittel und dessen Beziehungen die Ablaufbeziehungen zwischen diesen Elementen sind.[Hess1996].

1.2 Definition Prozessorientierung

- Abbildung der Vorgänge des realen Unternehmens in eine Sammlung miteinander kooperierender [Staud2001] bzw. nebeneinander wirkender Geschäftsprozesse.
- Daraus folgt die *Prozessintegration*:
Verbindung einzelner Prozesse oder Vorgänge[Mert1995] .

1.3 Definition Business Process Reengineering

- Oberbegriff für Methoden zur prozessorientierten Umgestaltung betrieblicher Organisationsstrukturen[Staud2001] .

- Optimierung von Betriebsvorgängen (aufgeteilt in mehr oder minder detaillierte Prozesse).

1.4 Ziele der GPM

- Dokumentation, Analyse und Gestaltung von Geschäftsprozessen [WIK], Bestandsaufnahme [Staud2001]
- Unterstützung der Kommunikation über Geschäftsprozesse [WIK]
- Unterstützung bei Auswahl und Einrichtung betriebswirtschaftlicher Standardsoftware [Staud2001, WIK]
- Qualitätsmanagement (Zertifizierung nach DIN ISO 9000 ff)[WIK, Staud2001]
- Business Process Reengineering[WIK] (s.o.).

1.5 Vorgehen:

- Aufteilung in Kernprozesse und unterstützende (support-) Prozesse [Staud2001]
 - Kernprozesse:
 - * wertschöpfend [Staud2001]
 - * Beitrag für den Unternehmenserfolg [Mert1995]
 - unterstützende Prozesse (supporting processes):
 - * wichtige Prozesse, ohne die die Kernprozesse nicht ablaufen können.
 - * Bsp. Laut [Steinbu1998]: Buchführung, Kostenrechnung, Lohn- und Gehaltsrechnung.

⇒In der Literatur nicht einheitlich definiert

⇒schlecht für Umsetzung.

1.5.1 Probleme:

- Aufteilung in Prozesse nicht einheitlich definiert bzw. definierbar (s.o.).
- Detailgrad der Modellierung (hier: Aufteilung in Prozesse) ebenfalls nicht einheitlich definiert bzw. prinzipbedingt *nicht definierbar*.

⇒Unschärfen bei der Betrachtung [Staud2001].

⇒Optimierungen sind dann nicht optimal.

1.6 Methodenübersicht zur Beschreibung von Prozessen

- **Unüblich:** Textliche Beschreibung.
 Vorteil: kann jeder verstehen.
 Nachteil: wird schnell unübersichtlich. Außerdem: aufwendig [Staud2001].
- **Verbreitet:** Beschreibung mittels Ereignisgesteuerter Prozessketten (EPK)
 Entwickelt 1992 von AUGUST-WILHELM SCHEER.
 Methode der eEPK (erweiterte EPK) nach [Scheer1997] in Deutschland weit verbreitet.
 eEPK sind eine vereinfachte Variante der Petrinetze (s.u.),
 Vorteil: leicht zu verstehen, tendieren aber mit steigender Beschreibungstiefe zur Unübersichtlichkeit.
 Nachteil: semi-formal. Mathematisch nicht 100%ig fundiert. Lässt Spielraum für Interpretationen. Umsetzung softwaretechnisch aufgrund des Semi-Formalismus nicht 1:1 möglich.
(Anm. SK: Verbreitet heißt also nicht: beste aller möglichen Lösungen.)
- **Besser für Simulationen:** Petrinetze (übliche Methode bei komplexen Prozessabläufen, z.B. UMTS-Netzplanung in DE).
 Entwickelt 1962 von CARL ADAM PETRI (* 1926), heute Professor an der Universität Hamburg.
 Vorteil: Petrinetze sind mathematisch fundiert und sehr genaue und komplexe Modellierungen (im Rahmen der generell auftauchenden Probleme, s.o.) sind möglich.
 Erweiterungen der Petrinetze → Höhere Petrinetze → VIPtools [Erwin2002].
 Nachteil: verlangen umfangreiches Hintergrundwissen und Einarbeitung.
 Bei steigender Beschreibungstiefe neigen sie, wie die EPKs, zur Unübersichtlichkeit.
- **Im Kommen:**UML (Unified Modeling Language)
 Entwickelt von GRADY BOOCH, JIM RUMBAUGH und IVAR JACOBSON.
 Vorteil: herstellerunabhängig.
 Ziel: Beschreibungslücke zwischen Geschäftsprozessmodellierung und Softwaretechnik eliminieren [WIK].
 Es handelt sich um einen objektorientierten Ansatz.
 Kommt aus der Softwareentwicklung, wo die objektorientierte Beschreibung und Programmierung schon seit Jahren erprobt und erfolgreich ist.

Kapitel 2

Modellierungsziel: Prozessoptimierung

2.1 Phase 1: Prozessanalyse

- auch genannt: Konzeptionelle Phase lt. [Staud2001].
- Intensive Diskussion zwischen den Prozessverantwortlichen/-mitwirkenden und den Modellierern [Staud2001]. Ziel soll sein, möglichst viele Informationen zur späteren Prozessmodellierung zu erlangen [Koch2001].
 - Im Gespräch unbedingt angesprochen werden müssen (lt. [Staud2001]):
 - * Abgrenzung des Geschäftsprozesses
 - * Feststellung der beteiligten Personen und ihrer Organisationseinheiten.
 - * Feststellung der beteiligten Informationsobjekte.
 - Notwendige Voraussetzungen sind lt. [Koch2001] weiterhin:
 - * Verständnis des Problemgebietes (beim Modellierer?)
 - * Zugang/Unterstützung durch die Mitarbeiter.
 - * Kenntnis der Modellierungstechnik.
 - Notwendige Informationen sind lt. [Koch2001]:
 - * Schritte des Prozesses und ihre Abfolge.
 - * Auftrittshäufigkeit des Prozesses.
 - * Zeiten der einzelnen Schritte (idealerweise Verteilung).

⇒ Anschließend: Textliche Beschreibung des gesamten Prozessablaufes. [Staud2001]
Selbstaufschreibung Arbeitsbericht [Koch2001].

In der Praxis wird die Phase der Prozessanalyse oft vernachlässigt ⇒ Fehler bei der Modellierung sind oft die Folge.

2.2 Phase 2: Optimierungsziel festlegen

- Im Gespräch mit den Prozessverantwortlichen/-mitwirkenden ist das Ziel der Optimierung festzulegen (Kostensenkung, Effizienzerhöhung, etc.).

Zu beachten ist, dass erst nach Festlegung des Optimierungsziels eine geeignete Methode zur Prozessmodellierung gewählt werden kann (so ist es beispielsweise mit eEPKs nicht ohne weiteres möglich, Zeitverläufe darzustellen und zu optimieren → eEPKs ungeeignet für Effizienzverbesserungen im Sinne von Durchsatzsteigerungen.)

2.3 Phase 3: Prozessmodellierung

- Anhand des in Phase 2 festgelegten Optimierungsziel ist eine Methode zur Prozessmodellierung zu bestimmen.

Geht es z.B. um die Optimierung von Ablaufbeziehungen so bieten sich Petrinetze und eEPKs an.

Geht es dagegen eher um die Optimierung hinsichtlich der Entwicklung einer neuen prozessunterstützenden oder -basierenden Software, so bietet sich der Einsatz der UML an.

2.4 Phase 4: Optimierung

- Optimierungen am Modell können mit verschiedenen Optimierungsmethoden durchgeführt werden.

Bei grafischen Modellen (Petrinetze, eEPK) können ggf. Redundanzen und überflüssige Handlungswege leicht erkannt und bereinigt werden.

Kapitel 3

Komponenten zur Prozessbeschreibung

In diesem Kapitel werden die drei momentan hauptsächlich verwendeten Methoden zur Prozessbeschreibung genauer beleuchtet.

Zu jeder Methode gibt es ein, mit den anderen direkt vergleichbares, Beispiel. Zusätzlich findet sich im Anhang A.1 auf Seite 19 noch eine Gegenüberstellung der drei Beispiele auf einer Seite.

3.1 Ereignisgesteuerte Prozessketten (EPK)

3.1.1 Funktionen

allumfassende Darstellung...

- zu leistende Tätigkeiten: das, was gemacht werden soll [Staud2001]
- zeitverbrauchendes Geschehen, das durch ein Startereignis ausgelöst und durch ein Endereignis abgeschlossen wird. [Scheer1998].
- Prinzipiell ist die Definition von Funktionen bzw. die Einteilung einzelner Arbeitsschritte in Funktionen dem Modellierer überlassen. Grundsätzlich sollte die Aufteilung in Funktionen allerdings nicht bis ins kleinste Detail vorgenommen werden sondern maximal bis zur letzten betriebswirtschaftlich sinnvollen Tätigkeit [Staud2001].
- Zeitverbrauch von Funktionen wird nicht erfasst [Staud2001].
(Anm. SK: in der mir vorliegenden Version 6.2 des ARIS Toolset können sowohl Zeiten als auch Kosten der jeweiligen Funktion erfasst werden).

3.1.2 Ereignisse

...versus...

- betriebswirtschaftlich relevante Ereignisse (z.B. Auftrag eingetroffen, Überweisung ist vorbereitet, Kundenanfrage ist abgelehnt) [Staud2001].
- Ein Ereignis ist ein Ergebnis einer Funktion.

- Einem Ereignis folgt wieder eine Funktion.
- Definition von Ereignissen ähnlich willkürlich wählbar wie bei Funktionen (s.o.).
- Bei Geschäftsprozessen gibt es in jedem Fall ein Startereignis und ein Endergebnis [Staud2001].

3.1.3 Organisationseinheiten

- werden Funktionen zugeordnet.
- Dienen der Festlegung, wo die in einer Funktion erfasste Aufgabe getätigt wird [Staud2001].
- Beispiele: Vertrieb, Personalwesen, Werk, etc. [Staud2001].

3.1.4 Informationsobjekte

- dienen zur Modellierung von Datenbeständen.
- sind an Funktionen geknüpft. Funktionen benötigen Informationen aus Informationsobjekten [Staud2001].
- Informationsobjekte erzeugen aber auch welche (Anm.: Informationen) [Staud2001].

3.1.5 Verknüpfung durch Operatoren und Kontrollfluss

- prinzipiell können ausschließlich Funktionen mit Funktionen und Ereignisse mit Ereignissen verknüpft werden.
- Die Verknüpfung erfolgt mit drei Operatoren: UND, ODER, XOR
- Der Kontrollfluss wird aus Übersichtsgründen von oben nach unten angeordnet.

3.1.6 Software

- ARIS Toolset (windowsbasiert, kommerzielle Lösung)
<http://www.ids-scheer.de>

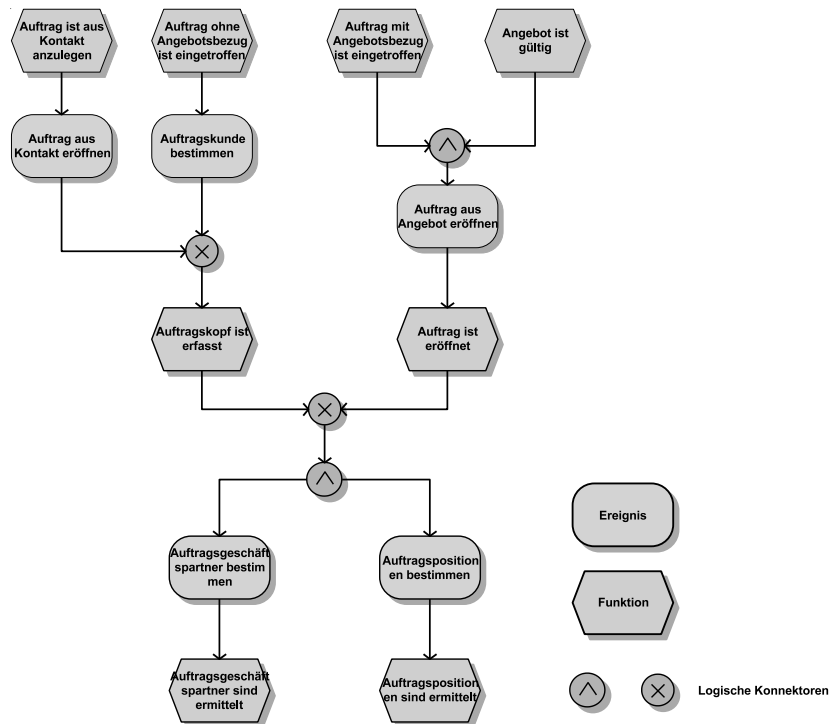


Abbildung 3.1: Beispiel für eine eEPK. Erstellt mit *ARIS Toolset 6.2* (Beispiel-Geschäftsfall aus [Moldt2000])

3.2 Petrinetze

...nüchterner Formalismus.

3.2.1 Stellen

- können Ereignisse oder Bedingungen sein.
- Eine markierte Stelle bezeichnet das Eintreffen des Ereignisses bzw. der Bedingung.
- Die Markierung erfolgt mit einer sog. Marke (engl. Token).
- Graphische Darstellung durch Kreis:
 unausgefüllt: ○ (Bedingung / Ereignis nicht eingetreten)
 ausgefüllt: ⊙ (Bedingung / Ereignis eingetreten).
- Bei S/T (Stellen/Transitions-Netzen) ist die Kapazität der Stellen ist zu definieren.
 Bei B/E (Bedingungs/Ereignis-Netzen) können:
 - Stellen nur eine Marke enthalten
 - Transitionen nehmen nur eine Marke weg.

3.2.2 Transitionen

- Was die “Funktionen” bei den EPKs sind, sind die Transitionen bei Petrinetzen.
- Es sind Aktivitäten, die bei gewissen Situationen¹ durchgeführt werden.
- Graphisch werden sie durch eine Linie oder ein Rechteck beschrieben: \square

3.2.3 Kanten

- Verbindet die einzelnen Elemente mittels Flusslinie.
- Bei S/T-Netzen können die Kanten gewichtet werden (wieviel Marken sie beim Schalten wegnehmen oder hinzufügen)
- Graphisch: \longrightarrow

3.2.4 Token

- stellen die Markierungen dar.
- werden durch das Netz weitergereicht (wandern).
- Eine Transition kann erst dann schalten, wenn auf allen ihren Eingängen mind. ein Token anliegt und an ihren Ausgängen Stellen mit ausreichend freier Kapazität zur Aufnahme der Token vorhanden ist.
- Graphisch werden die Token durch Punkte in den Stellen dargestellt (\odot), oder aber (bei größeren Mengen von Token) durch eine Ziffer für die Anzahl der Markierungen.

3.2.5 Software

- winpetri (windowsbasiert, frei verfügbar)
<http://fresh.t-systems-sfr.com/pc/src/misc/winpetri.zip>
- renew “The Reference Net Workshop” (javabasiert, frei verfügbar)
<http://www.renew.de>
- VIPTool (javabasiert, frei verfügbar)
<http://www.informatik.ku-eichstaett.de/projekte/vip/viptool/viptool.html>
- Liste mit weiteren Petrinetz-Tools (umfassend):
Petri Nets World: All Records in Petri Nets Tools Database http://www.daimi.au.dk/PetriNets/tools/complete_db.html

¹bei Erfüllung einer Voraussetzung

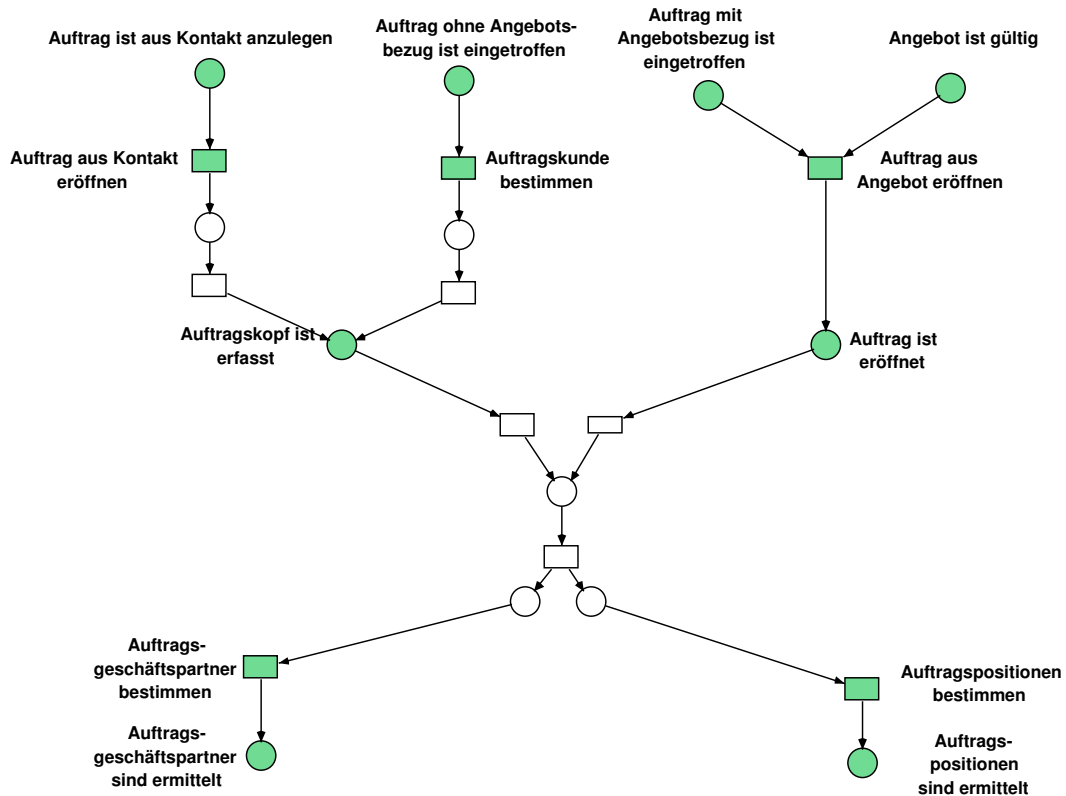


Abbildung 3.2: Beispiel für ein Petrinetz. Erstellt mit der Modellierungssoftware *renew*. (Beispielgeschäftsfall aus [Moldt2000])

3.3 UML Aktivitätsdiagramme

Das Beste aus beiden Welten?

Eine der (derzeit) acht Diagrammtypen der UML stellt das *Aktivitätsdiagramm* dar. Dieser Diagrammtyp ist eng verwandt mit Programmablaufplänen (sog. Flussdiagrammen) [GräBau2004]. Auch finden sich Elemente der Petrinetze wieder (Transitionen, Start- und Endzustand²).

Demnach ist die Vergleichbarkeit zu den in 3.1 auf Seite 8 und 3.2 auf Seite 10 genannten Methoden durchaus gegeben.

3.3.1 Aktivität

“Eine Aktivität ist ein einzelner Schritt in einem Ablauf von Aktivitäten, die zusammen einen Geschäftsprozess bilden, der eine Leistung eines Geschäftssystems beschreibt.” [GräBau2004]

- Beim praktischen Modellieren hilft die Faustregel, dass man UML-Aktivitäten am ehesten mit EPK-Funktionen oder Petrinetz-Transitionen vergleichen kann.

²auf die hier nicht näher eingegangen wird.

3.3.2 Übergang (engl. hier: Transition)

- Auf den ersten Blick vergleichbar mit dem *Kontrollfluss* bei EPKs oder den *Kanten* bei Petrinetzen.
- Prinzipbedingt kann ein Übergang in der UML aber auch beschriftet werden, was ihn dann zu einer den EPK-*Ereignissen* oder den Petrinetz-*Stellen* vergleichbaren Instanz macht.

⇒ Verschmelzung des Kontrollfluss mit Ereignissen, die durch Aktivitäten ausgelöst wurden.

(Anm. SK: nicht optimal bei der Darstellung. Es mag aber formale Vorzüge haben.)

Hinweis: einen Ausweg aus dem Dilemma bilden die sog. Objektflusszustände (Object Flow State)

3.3.3 Verzweigung (Branch)

- Verzweigungspunkt für *Übergänge*.
- Hat *einen* Eingang und *zwei oder mehr* Ausgänge (Transitionen).
- Ausgänge können mit einer Wächterbedingung (boolescher Ausdruck) versehen werden.
- Graphische Darstellung: Raute ◇

3.3.4 Zusammenführung (Branch, Boolean: ODER)

In der Literatur liegen auseinandergehende Betrachtungsansätze vor. So definiert [Oest2003] eine Zusammenführung als “Disjunktion, ODER-Verknüpfung” und stellt ihn durch eine “*nichtausgefüllte Raute mit mehreren eingehenden und einer ausgehenden Transition*” dar.

In [GräBau2004] hingegen wird eine Zusammenführung durch einen Synchronisationsbalken dargestellt, der mehrere Transitionen als Eingänge hat und eine Transition als Ausgang hat.

In diesem Manuskript wird der Notation nach [Oest2003] der Vorzug gegeben, da sie logischer erscheint und reale Problemfälle besser abbildet.

- Zusammenführungspunkt für *Übergänge*.
- Hat mehrere Eingänge und einen Ausgang.
- Eingänge können mit einer Wächterbedingung (boolescher Ausdruck) versehen werden.
- Graphische Darstellung: Raute ◇

3.3.5 Teilung (Fork)

- Aufteilungspunkt für Übergänge.
- Hat einen Eingang und mehrere Ausgänge.
- Der Eingang teilt sich sofort - ohne Bedingung - in mehrere ausgehende nebenläufige Übergänge auf.
- Graphische Darstellung: Balken

3.3.6 Synchronisation (Join, Boolean: UND)

- Synchronisationsstelle für Übergänge.
- Hat mehrere Eingänge und einen Ausgang.
- Es wird auf alle eingehenden Übergänge gewartet, bevor der Kontrollfluss fortgesetzt wird, vgl. [Oest2003], S. 182.
- Graphische Darstellung: Balken

3.3.7 Objektknoten (Object Flow State)

- Repräsentiert ein Objekt oder eine Menge von Objekten innerhalb des Aktivitätsdiagramms, vgl. [Oest2003], S. 184.
- Graphische Darstellung: Rechteck, welches den Namen des Objektknotens und optional in eckigen Klammern dessen Zustand beinhaltet.

3.3.8 Software

- ArgoUML (javabasiert, frei verfügbar)
<http://argouml.tigris.org>
- Poseidon for UML (javabasiert, kommerzielle Lösung - allerdings ist eine Community Edition frei verfügbar)
<http://www.gentleware.com>
- Magic Draw UML (javabasiert, kommerzielle Lösung - allerdings ist eine Community Edition frei verfügbar)
<http://www.magicdraw.com>
- ARIS Toolset [eingeschränkt benutzbar] (windowsbasiert, kommerzielle Lösung)
<http://www.ids-scheer.de>

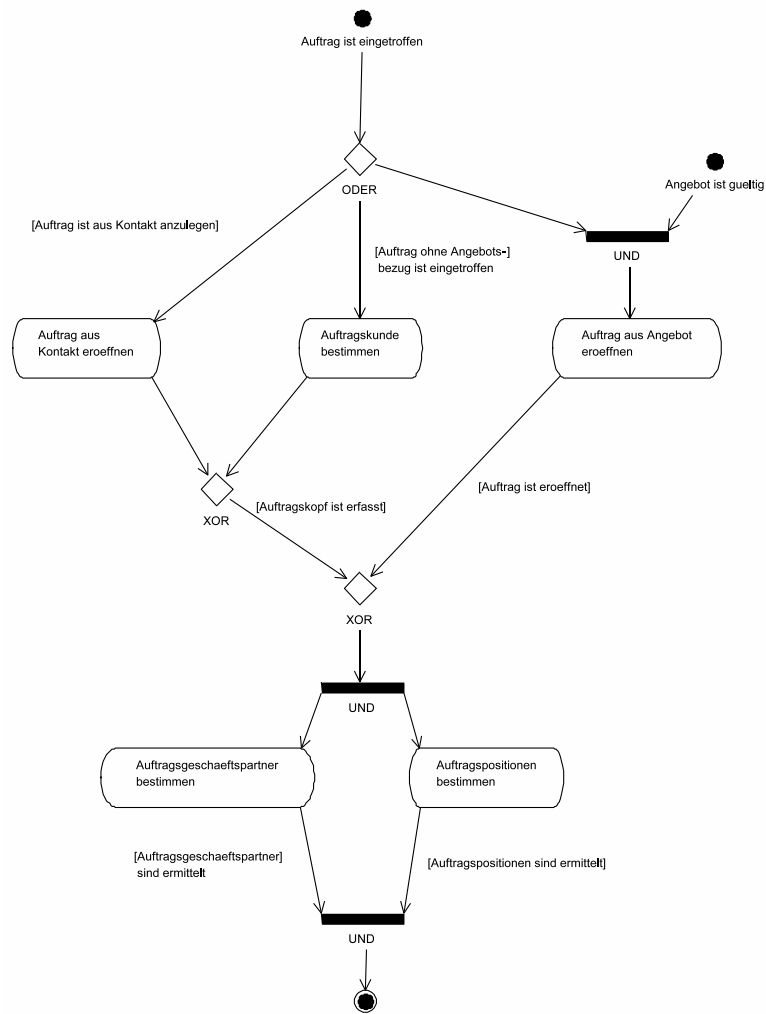


Abbildung 3.3: Beispiel für ein UML-Aktivitätsdiagramm. Erstellt mit der Software *Poseidon CE*.

Literaturverzeichnis

- [WIK] Freie Enzyklopädie Wikipedia: www.de.wikipedia.org
- [Staud2001] Staud, Josef: Geschäftsprozessanalyse (2. Auflage), Berlin u.a. 2001
- [HaCha1995] Hammer, Michael; Champy, James: Business Reengineering: Die Radikalkur für das Unternehmen, Frankfurt, New York 1995
- [KeTeu1997] Keller, Gerhart; Teufel, Thomas: SAP R/3 prozessorientiert anwenden. Iteratives Prozess-Prototyping zur Bildung von Wertschöpfungsketten, Bonn u.a. 1997
- [Hess1996] Hess,Thomas: Entwurf betrieblicher Prozesse. Grundlagen Bestehende Methoden Neue Ansätze, Wiesbaden 1996
- [Mert1995] Mertens, Peter: Integrierte Informationsverarbeitung 1. Administrations- und Dispositionssysteme in der Industrie (10. Auflage), Wiesbaden 1995
- [Steinbu1998] Steinbuch, Pitter A. (Hrsg.): Prozessorganisation Business Reengineering Beispiel R/3, Ludwigshafen (Rhein) 1998
- [Scheer1997] Scheer, August-Wilhelm: Wirtschaftsinformatik. Referenzmodelle für industrielle Geschäftsprozesse (7. Auflage), Berlin u.a. 1997
- [Scheer1998] Scheer, August-Wilhelm: ARIS vom Geschäftsprozess zum Anwendungssystem (3. Auflage), Berlin u.a. 1998
- [Erwin2002] Erwin, Thomas: Entwurf von Geschäftsprozessen mit Petrinetzen, Diss., <http://www.ubka.uni-karlsruhe.de/cgi-bin/psview?document=/2002/wiwi/7> (abgerufen am: 22.04.04), Karlsruhe 2002 dazu VIPtools Download: <http://www.informatik.ku-eichstaett.de/projekte/vip/> (abgerufen am: 22.04.04).
- [Koch2001] Koch, Stefan; Bernroider, Edward: Einführung in die Prozessmodellierung, Präsentation WU Wien 2001, <http://www.wi.wu-wien.ac.at/~koch/lehre/untmod-ag-ws-01/prozmod.pdf> (abgerufen am: 03.05.2004).

- [GräBau2004] Grässle, Patrick; Baumann, Henriette; Baumann, Philipe: UML projektorientiert, Bonn 2003
- [Moldt2000] Moldt, Daniel; Rodenhagen, Jörg: Ereignisgesteuerte Prozeßketten und Petrinetze zur Modellierung von Workflows, Studie, Hamburg 2000, <http://wwwmath.uni-muenster.de/cs/u/versys/workshops/VVNS2000/papers/Moldt.pdf>
- [Oest2003] Oestreich, Bernd; Weiss, Christian; Schröder, Claudia; Weilkiens, Tim; Lenhard, Alexander: Objektorientierte Geschäftsprozessmodellierung mit der UML, Heidelberg 2003
- [HeiSei2002] Seidlmeier, Heinrich: Prozessmodellierung mit ARIS, Braunschweig/Wiesbaden 2002
- [Learning UML] Si Alhir, Sinan: Learning UML, Sabastopol, CA 2003
- [MaHi2003] Hitz, Martin; Kappel, Gerti: UML@work - Von der Analyse zur Realisierung (2. Auflage), Heidelberg 2003
- [HeBa2000] Balzert, Heide: Objektorientierung in 7 Tagen - Vom UML-Modell zur fertigen Web-Anwendung, Heidelberg u.a. 2000

Anhang A

Anhang

A.1 Vergleich eEPK, Petrinetz und UML-Aktivitätsdiagramm

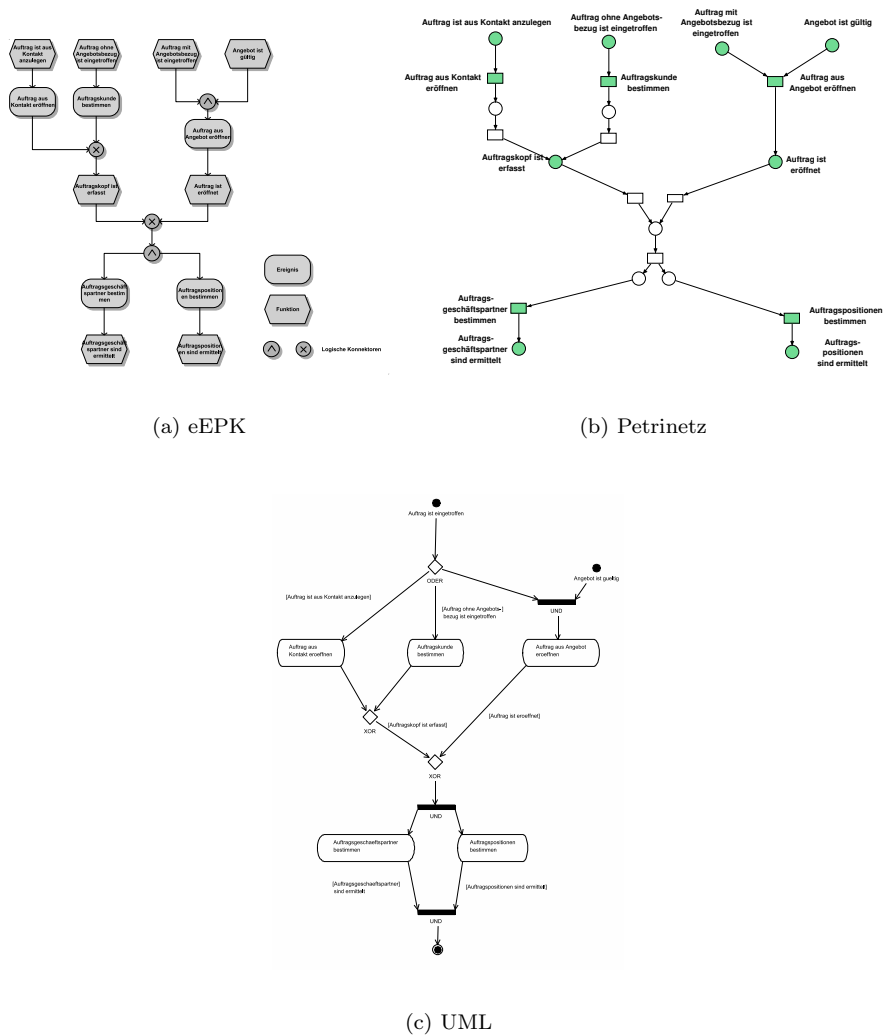


Abbildung A.1: Direkter Vergleich der drei Methoden