

Geschäftsprozess-Management / SCM

Prof. Dr.-Ing. Andreas Ittner

Email: ittner@hs-mittweida.de

WWW: www.hs-mittweida.de/~ittner

Tel.: +49(0)3727-58-1288

Mob.: +49(0)177-5555-347

Zu meiner Person

- 1990-1998 Informatik-Studium an der TH Leipzig und TU Chemnitz, Promotion, Schwerpunkt **KI und Data Mining (Big Data)**,
- 1998 **Gründung** der prudsys AG (ehem. Prudential Systems Software GmbH) aus der TU Chemnitz heraus, 1998-2006 Geschäftsführer/Vorstand, 2006-2009 Aufsichtsrat der **prudsys AG**,
- seit 2007 **Professor für Informatik/Verteilte Informationssysteme** an der Hochschule Mittweida, Forschergruppe mit mehreren Mitarbeitern,
- zahlreiche **FuE-Projekte mit Big-Data- und Blockchain-Fokus** sowie Betreuung von Projekt-/Bachelor-/Master-Arbeiten,
- seit Juni 2017 Leiter des **Blockchain Competence Center Mittweida** (<http://blockchain.hs-mittweida.de>)

Organisatorisches

- Vorlesung: 15 Veranstaltungen, lt. aktueller Stundenplanung,
- Praktika/Übungen, lt. Stundenplan,
- Folien (nach der LV) im Laufwerk R unter R:\CB\Ittner\GPM\,

- Klausur:
 - Bearbeitungszeit 90 Minuten.



- Kontakt:
 - Prof. Dr.-Ing. Andreas Ittner
 - ittner@hs-mittweida.de, Durchwahl -1288
 - Raum 8-312

Literatur / weiterführende Unterlagen

- Richter-v.Hagen, Stucky Business-Process- und Workflow-Management; Teubner, 2004.
- Freund, J., Rücker, B.: „Praxishandbuch BPMN 2.0“, Hanser Verlag 2010, ISBN 978-3-446-42455-5.
- Allweyer, Th.: „BPMN 2.0 Business Process Model and Notation - Einführung in den Standard für die Geschäftsprozessmodellierung“, ISBN 978-3-8391-2134-4.
- Aalst, W. v. d. The Application of Petri Nets to Workflow Management,
- Spezifikationsdokument zur BPMN: www.omg.org/spec/BPMN/
- Workflow-Management Coalition (Terminology, Glossary, Reference Model) <http://www.wfmc.org>
- Sinnbilder für Datenfluss- und Programmablaufpläne DIN 66001.

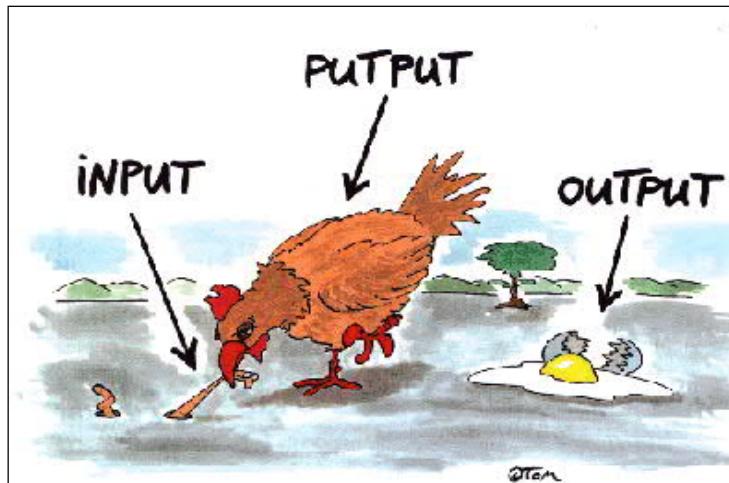
Gliederung

- Motivation / Bezug zum ECM
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs und BPMN
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards
- Methoden der Prozessverbesserung
- GPs in der Praxis
- Prozess-Controlling
- Prozess-Qualität und -Kosten

Gliederung

- Motivation / Bezug zum ECM
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs und BPMN
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards
- Methoden der Prozessverbesserung
- GPs in der Praxis
- Prozess-Controlling
- Prozess-Qualität und -Kosten

Ein praxistypischer „Produktionsprozess“



7

06.10.2017

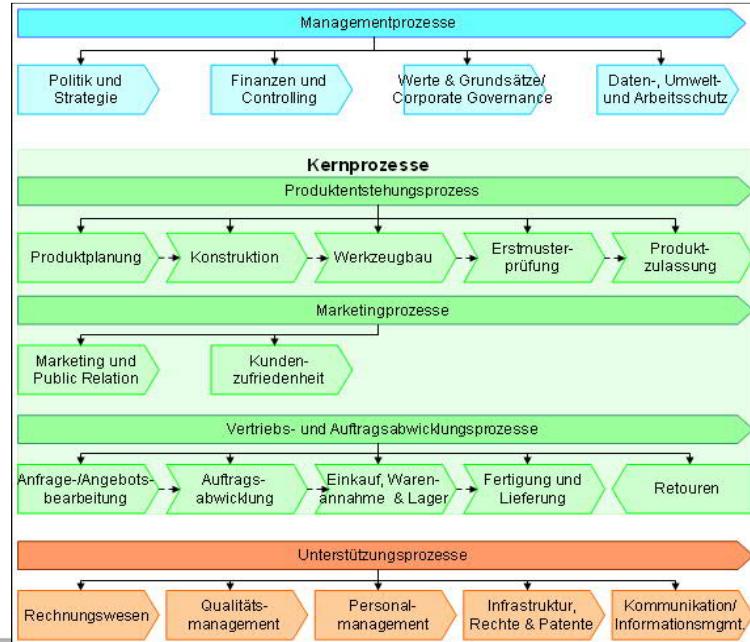
Warum Geschäftsprozess-Management?

- Ein Geschäftsprozess ist eine Folge von Schritten oder ein „Rezept“, um ein Geschäftsresultat zu erzielen.
- Im Gegensatz zum Projekt wird der Prozess öfter durchlaufen.
- Ein Geschäftsprozess kann Teil eines anderen Geschäftsprozesses sein oder andere Geschäftsprozesse enthalten bzw. diese anstoßen.
- Geschäftsprozesse gehen oft über Abteilungen und Betriebsgrenzen hinweg.
- GPs gehören zur Ablauforganisation eines Unternehmens.
- Ein gutes Geschäftsprozess-Management dient dazu, um hohe Effektivität und Effizienz bei den Geschäftsprozessen zu erzielen.

8

06.10.2017

Beispiele für Geschäftsprozesse

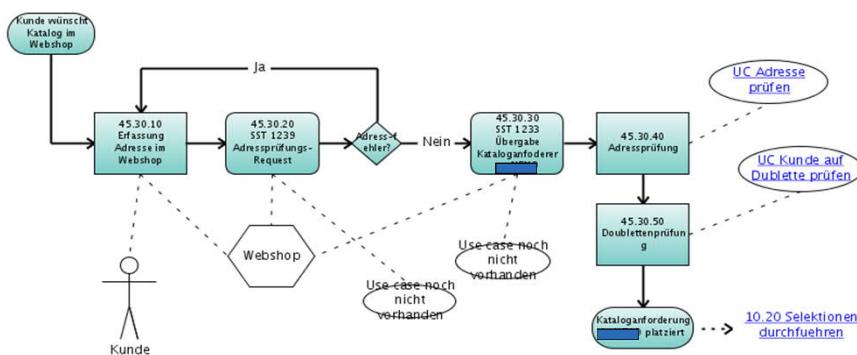


9

06.10.2017

Beispiel-Prozess mit Gliffy (www.gliffy.com)

- Beispiel-Prozess im Versandhandel: „Kataloganforderung im Webshop“

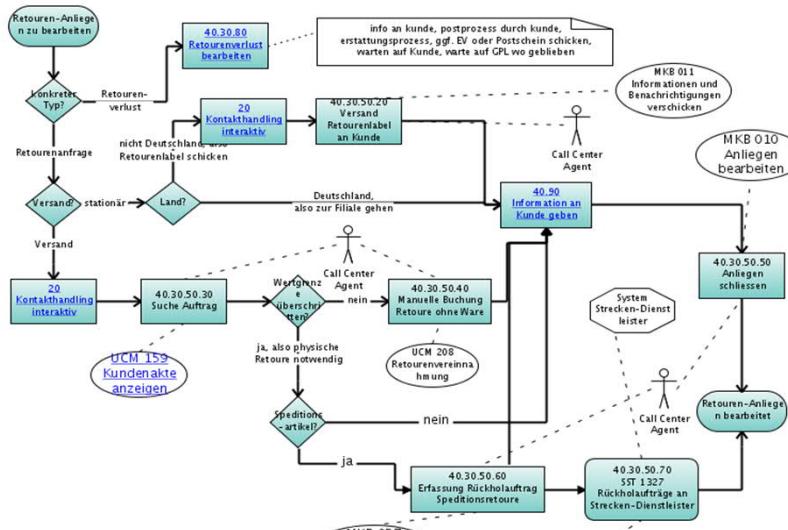


10

06.10.2017

Beispiel-Prozess mit Gliffy (www.gliffy.com)

- Beispiel-Prozess im Versandhandel: „Retourenanliegen bearbeiten“

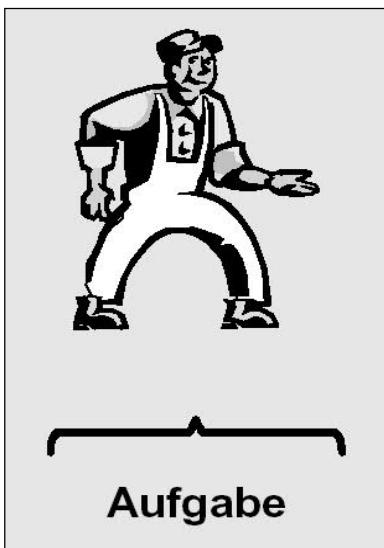


11

06.10.2017

Motivation und Einführung

- Arbeitsteilung und ihre Folgen

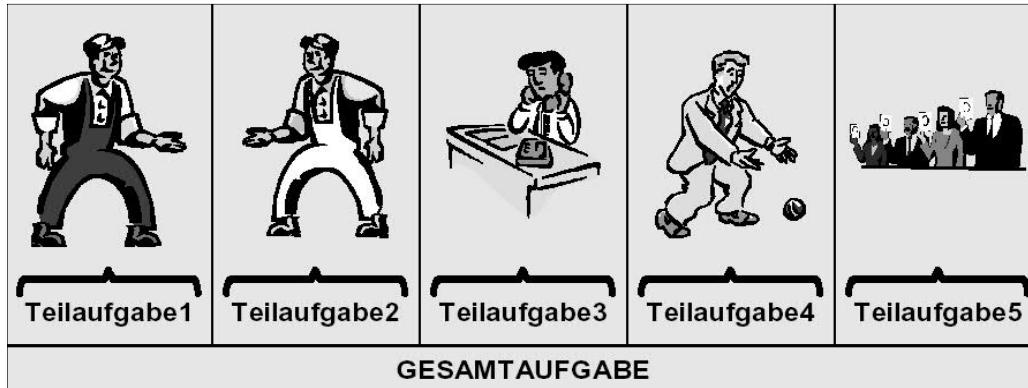


Vorindustrielle Zeit

- Keine/geringe Arbeitsteilung (d.h. alle Aufgaben von einer Person ausgeführt),
- Teilaufgabe nicht Selbstzweck sondern Teil der Gesamtaufgabe,
- „automatische“ Verbesserung des Gesamtablaufs im Kopf der jeweiligen Person.

Motivation und Einführung

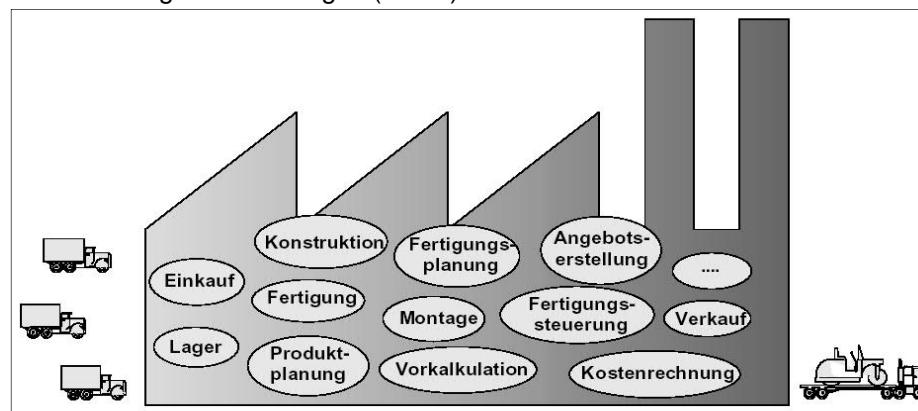
- Arbeitsteilung und ihre Folgen (Forts.)



- Arbeitsteiliger Prozess
 - Aufteilung der Gesamtaufgabe in Teilaufgaben,
 - Bearbeitung der Teilaufgaben durch **verschiedene** Personen/Funktionen.

Motivation und Einführung

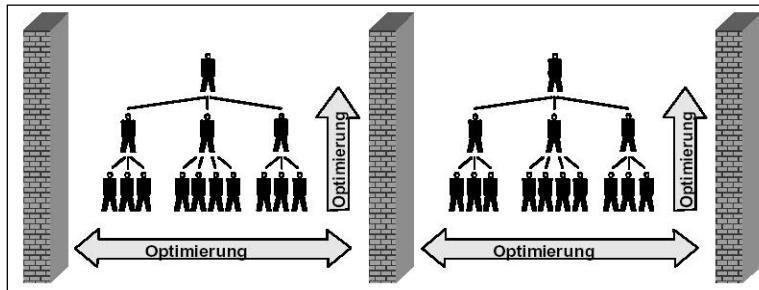
- Arbeitsteilung und ihre Folgen (Forts.)



- Industrielle Fertigung → arbeitsteiliger Prozess.

Motivation und Einführung

- Arbeitsteilung und ihre Folgen (Forts.)



- Auftretende Probleme/Herausforderungen:
 - Zusammenhang zwischen Teil- und Gesamtaufgabe für den Einzelnen nicht mehr erkennbar,
 - Optimierung endet an den Abteilungs-/Bereichsgrenzen,
 - einsetzende „vertikale“ Optimierung,
 - Gefahr: Teilaufgabe/-funktion wird zum Selbstzweck.

Motivation und Einführung

- Nachteile der Arbeitsteilung

- Fehler durch Medienbrüche, Organisationsbrüche,
- Verbesserungsvorschläge / organisatorische Anpassungen beschränken sich i.d.R. auf einzelne Arbeitsschritte,
- nur selten Analyse des übergeordneten Zusammenhangs,
- Doppelarbeiten (aufgrund nicht auffindbarer Daten),
- Redundanz, mangelnde Aktualität und Inkonsistenz (aufgrund Mehrfacharchivierung),
- Schlechte Informationsbereitstellung für Kunden (aufgrund erheblicher Zugriffszeiten),
- Lange Durchlaufzeiten von Vorgängen (aufgrund sehr starker Arbeitsteilung),
- Verschärfung dieser Problematik im Rahmen der Globalisierung.

Motivation und Einführung

- Wirtschaftliche Aspekte:
 - Aufgeblähte, undurchsichtige Abläufe, d.h. Unternehmen werden unbeweglich und teuer,
 - Internationalisierung/Globalisierung der Märkte, d.h. enorm wachsender Wettbewerbsdruck,
- Verbesserung der Effizienz (Kosten/Zeit/Ressourcen) wird zur Überlebensfrage!
- Erwünschter, angestrebter Zustand:
 - Leistungsfähige Organisationsstrukturen, die Geschäftsabläufe effizient abarbeiten und sich gleichzeitig wandelnden Anforderungen anpassen können.

Motivation und Einführung

- Konkrete Ziele:
 - Verschlankung der Geschäftsprozesse/Entscheidungswege,
 - Verkürzung der Entwicklungszeiten („time to market“),
 - Verkürzung der Fertigungszeiten (bei gleich hoher Qualität),
 - Rasche Reaktion auf Marktveränderungen.

→ ***implizieren auch ständige Anpassung der Geschäftsprozesse, Business Process Re-Engineering.***
- Weg:
 - vernachlässigte Gesamtablauforientierung berücksichtigen, d.h. Gesamtablaufmodellierung, -analyse und –ausführung,
 - **Forderung:** Orientierung der betrieblichen IT am Gesamtablauf,
 - **Wunsch:** durchgängige, vorgangsorientierte IT-Unterstützung für den gesamten Geschäftsablauf.

Motivation und Einführung

- Durchgängige, vorgangsorientierte IT-Unterstützung für alle Geschäftsprozesse? Was ist damit gemeint?



Workflow-Management am Beispiel einer Tagesklinik

- Ambulanz-Aufnahme**

Ambulanzkraft (KA)

- Überweisungen entgegennehmen
- administrative Aufnahmen
- Entlassungen

- Ambulanz-Arzt**

Arzt (AA), Oberarzt

- Patientin untersuchen
- Aufklärung,
- Untersuchungen anordnen,
- Durchsicht der Patientenakte,
- Überprüfung von Untersuchungsergebnissen,
- Indikation stellen/absegnen.

- Ambulanz-Anästhesie**

Anästhesist (extern)

- OP-Bereich**

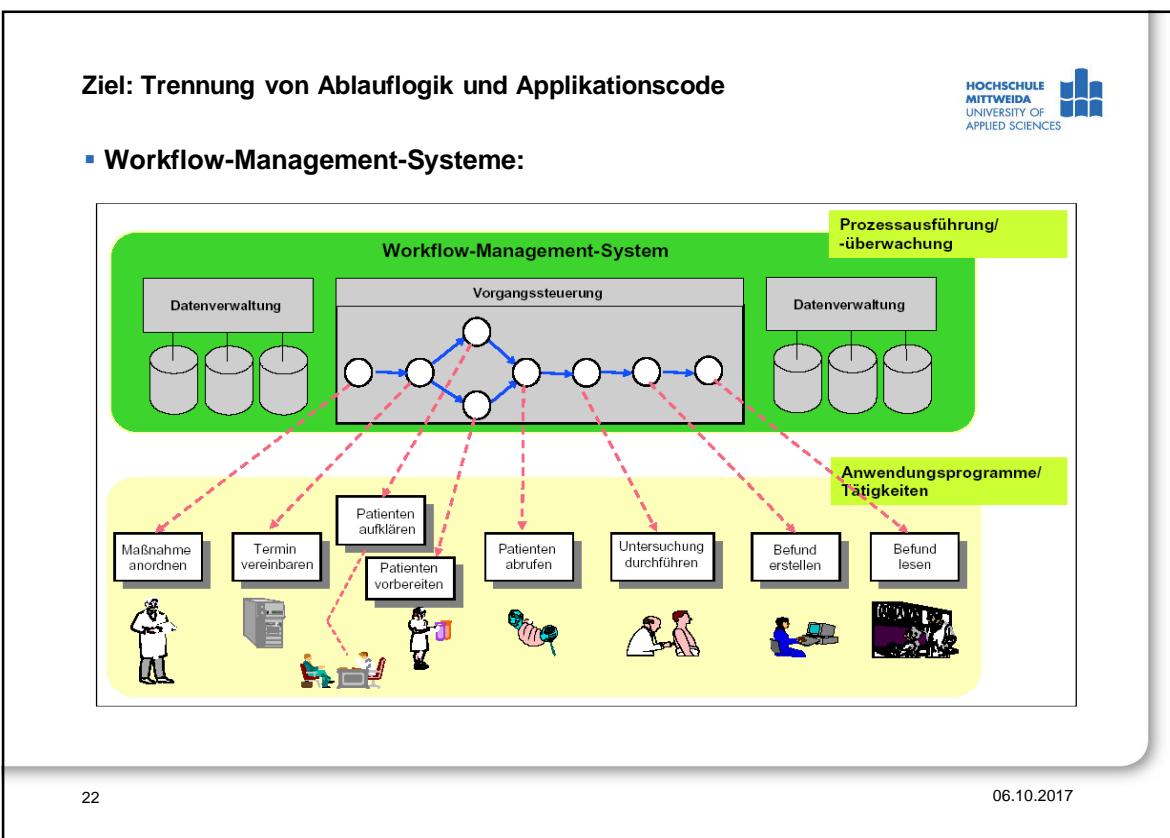
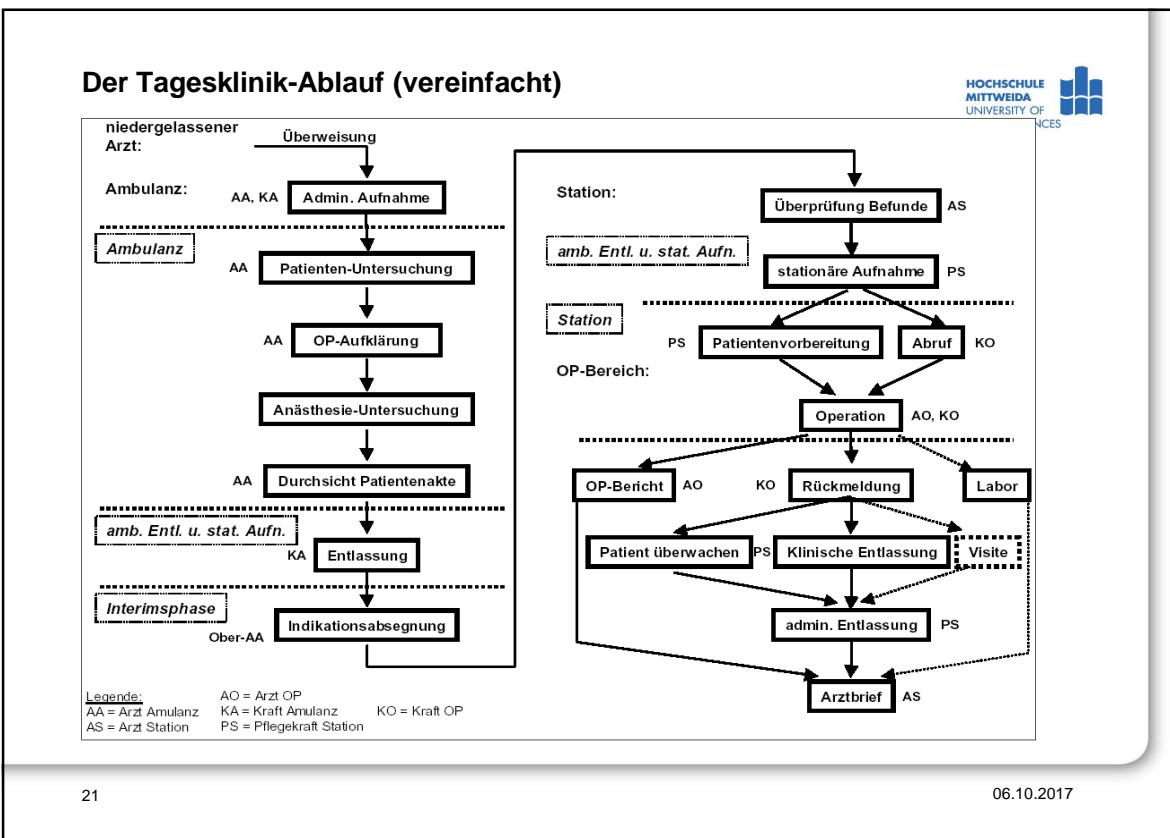
Arzt (AO), OP-Planung (KO), (OP-Team), (OP-Kraft)

- OP-Dokumentation, -Bericht (AO),
- Aufstellen des OP-Planes (KO),
- Abruf, Rückmeldung (KO).

- Station**

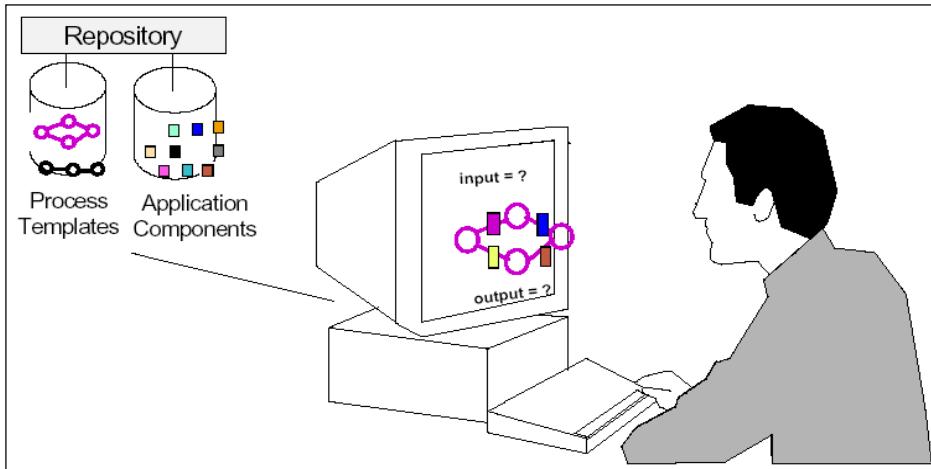
Stationskraft (PS) Arzt (AS)

- Pflegedokumentation (PS)
- Administrative Aufnahme/Entlassung (PS),
- Arztbrief (AS),
- Patientenakten einsehen (AS),
- Klinische Entlassungen (AS).



Vision zukünftige Anwendungsentwicklung

- Komposition und Wartung vorgangsorientierter Informations- und Anwendungssysteme aus Prozessvorlagen und Anwendungskomponenten



23

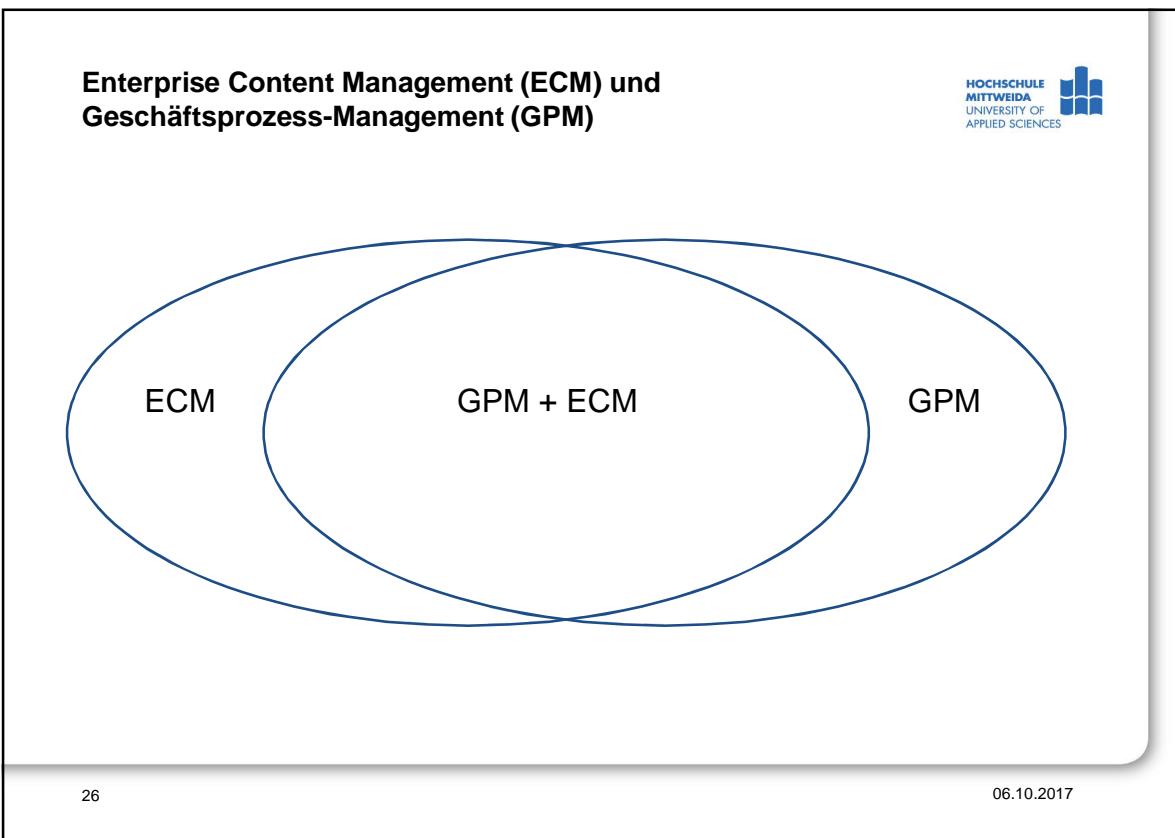
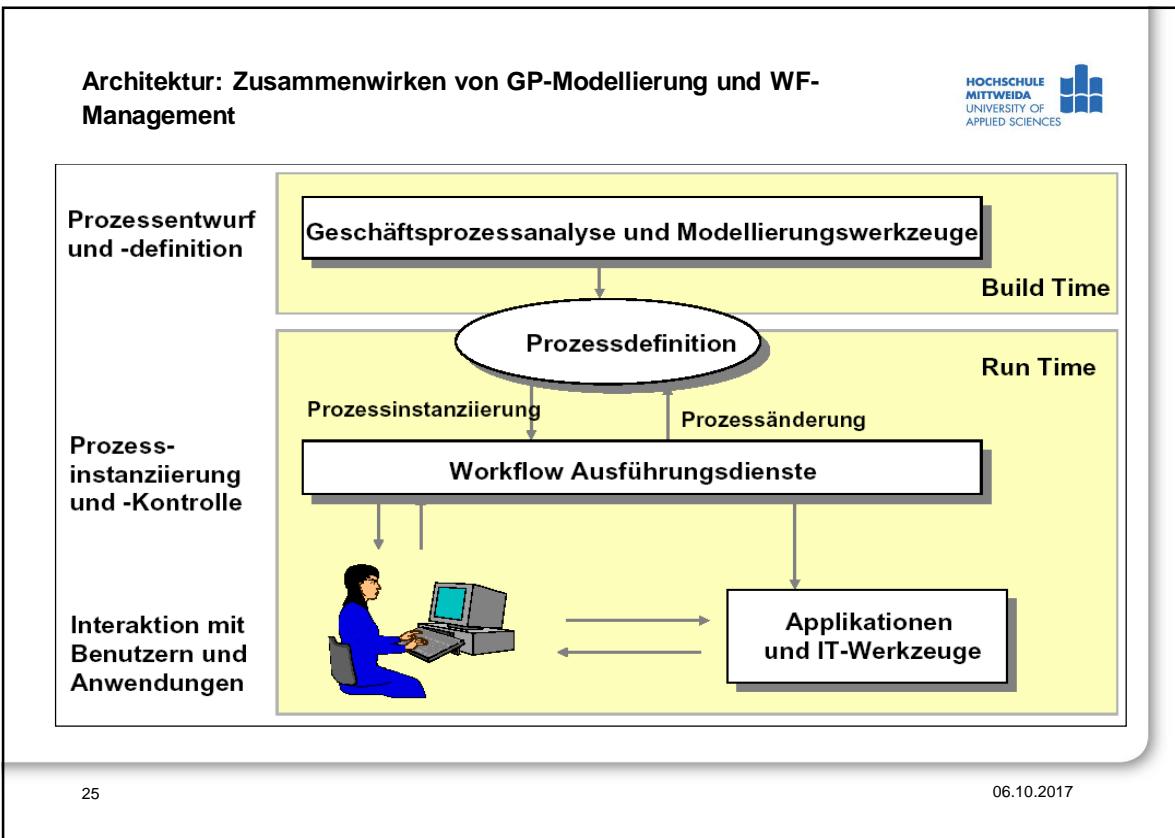
06.10.2017

Realisierung vorgangsorientierter Anwendungssysteme mittels (prozessorientiertem) Workflow-Management-System

- **Positiv:**
 - Explizite Modellierung der Abläufe,
 - dadurch einfachere Anpassung an Änderungen,
 - visualisierte Animation hilft frühzeitig Fehler im Ablauf zu erkennen,
 - Potenzial für systemseitige Selbstüberwachung und Fehlerbehandlung,
 - damit Entlastung des Anwendungsentwicklers von systemnahen Aspekten.
- **Jedoch:**
 - heutige Systeme z.T. noch sehr eingeschränkt (z.B. zu unflexibel),
 - große Unterschiede in der angebotenen Funktionalität.

24

06.10.2017



Enterprise Content Management (ECM) und GPM

- Youtube Video: [Die Wahrheit über ECM](#)

Allgemeine ECM-Aufgaben:

- Erfassung
- Verwaltung
- Speicherung
- Aufbewahrung
- Bereitstellung



27

06.10.2017

Enterprise Content Management (ECM) und GPM

Enterprise Content Management (ECM) kann angesehen werden als

1. Strategischer Ansatz, (nahezu beliebige) Informationen zu verwalten und
2. als Softwarelösung / Software Toolset.

Als **strategischer Ansatz** kann ECM einem Unternehmen helfen, seine Informationen (Content) so zu verwalten / zu managen, dass die Effizienz und Effektivität gesteigert, Kollaboration unterstützt und das Teilen von Information erleichtert wird.

Als **Software(toolset)** besteht ECM aus einer Menge von Komponenten und Applikationen für das Content Life Cycle Management, die miteinander wirken oder aber auch als Stand-alone-Lösung eingesetzt werden können.

28

06.10.2017

Enterprise Content Management (ECM) und GPM



29

06.10.2017

Enterprise Content Management (ECM) und GPM

Kern-Komponenten einer ECM-Lösung sind Folgende:

- Document Management (DM),
- Image-processing Applications,
- Workflow/Business Process Management (BPM),
- Records Management (RM),
- Web Content Management (WCM),
- Social Content
 - Document Sharing,
 - Collaboration,
 - Knowledge Management,
- Extended Components
 - Digital Asset Management (DAM),
 - Document Composition,
 - E-Forms,
 - Search,
 - Content and Analytics,
 - Email and Information Archiving,
 - Email Management,
 - Packaged Application Integration.



30

06.10.2017

Document Management (DM)

- Kann als **Kern (Core)** einer ECM-Lösungen betrachtet werden,
- Stellt Funktionalitäten für
 - das Check-in / Check-out von Dokumenten zur Verfügung,
 - eine Versionsverwaltung bereit,
 - Sicherheits- und Bibliotheksservices für Geschäftsdokumente zur Verfügung.
- Darüber hinaus sind bei manchen Lösungen auch Komponenten für
 - das Zusammenführen / Verbinden von Dokumenten (Compound Document Support) sowie
 - für das Replizieren von Informationen (Content Replication) im Einsatz.

Image-processing Applications

- dienen dem Erfassen, Verarbeiten und Verwalten von Bildern (Images) von papiergebundenen Dokumenten,
- Erfassung:
 - Hard- und Software zum Scannen,
 - Optical Character Recognition (OCR),
 - Formgebundene Verarbeitung (Formulare).
- Diese Funktionalitäten werden entweder selbst oder über 3rd-Party Lösungen zur Verfügung gestellt.
- Speicherung von erfassten (eingescannten) Dokumenten in einem oder mehreren Repositories (Datenbanken, File Systeme) sowie deren Routing in entsprechenden Prozessen.

Workflow/Business Process Management (BPM)

- Zur Unterstützung von Geschäftsprozessen,
- Zum Routing von Content / Informationen,
- Zur Zuordnung von Arbeitsaufgaben (Tasks) und Zuständen (States),
- Zur Prüfung, d.h. Nachverfolgbarkeit und Sicherheit (Audit Trails).

- Minimale Anforderungen sind die Unterstützung einer Dokumenten-Überprüfung (Review) sowie von Genehmigungs-Prozessen (Approval Workflow).
- Des weiteren haben manche Lösungen auch Komponenten zur grafischen Prozess-Erstellung/-Verwaltung sowie eine Unterstützung von seriellen und parallelen Abläufen.
- Grad des Ausbaus eines BPM in einem ECM kann sehr unterschiedlich sein.

Records Management (RM)

- Automatismen und Regeln für eine langfristige Aufbewahrung von Informationen (Content) entsprechend den gesetzlichen, regulativen und Branchen-/Industrie-spezifischen Anforderungen.

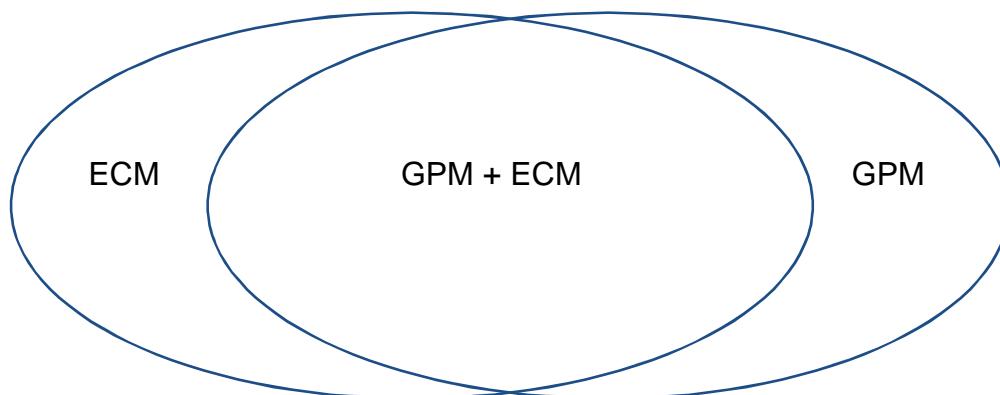
- Minimale Anforderungen besteht in einer Aufbewahrung von geschäftskritischen Informationen gemäß eines vorgegebenen Archivierungsplanes (records retention schedule).

- Die Anforderungen an ein RM können je nach Branche und Einsatzgebiet sehr stark variieren.

Web Content Management (WCM)

- Für die webbasierte Steuerung und die Beeinflussung von Content (Intra- und/oder Internet),
 - Erzeugung von Content via Templates,
 - Workflow- und Änderungsmanagement (z.B. Freigaben, Versionierung, etc.),
 - Deployment von Content, d.h. Lieferung von vorbereitetem oder on-demand Content auf Web-Server,
-
- An ein ECM anzubindende WCM-Komponenten werden meist über 3rd-Party-Lösungen zur Verfügung gestellt,
 - das Spektrum dieser Lösungen ist sehr breit,
 - anspruchsvolle Lösungen adressieren auch Aspekte wie z.B. Replikation, Last-Verteilung oder auch Web-Analytics.

Enterprise Content Management (ECM) und Geschäftsprozess-Management (GPM)



Zusammenfassung

- Verbesserung der Effizienz überlebenswichtig für (fast) alle Unternehmen,
- Verstehen der betrieblichen Prozesse wesentliche Voraussetzung zur Effizienzsteigerung,
- Statt: „Organize before automate“: „process thinking“,
- Moderne Modellierungswerkzeuge können hierbei entscheidend helfen,
- Ideale Kombination: Unterstützung der optimierten Prozesse durch vorgangsorientierte (prozessorientierte) betriebliche Informationssysteme,
- Allerdings: „Klassische“ Art der Anwendungsentwicklung / Informationsbereitstellung stößt hierbei an ihre Grenzen,
- Workflow-Management-Systeme können helfen, diese Grenzen zu überwinden,
- Enterprise Content Management (ECM) spielt eine entscheidende Rolle!,
- **Aber:** Möglichkeiten und Grenzen der Systeme müssen realistisch eingeschätzt werden!
- **Paradigmenwechsel:**
 - von informationszentriert zu prozesszentriert
 - Früher: Informationssysteme unterstützen die Ausführung einzelner Aufgaben,
 - Heute: Informationssysteme unterstützen Geschäftsprozesse, „they manage the flow of work“

Geschäftsprozess-Management / SCM

Prof. Dr.-Ing. Andreas Ittner

Email: ittner@hs-mittweida.de

WWW: www.hs-mittweida.de/~ittner

Tel.: +49(0)3727-58-1288

Mob.: +49(0)177-5555-347

Gliederung

- Motivation
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs, BPMN, ...
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards

- Motivation
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs, BPMN, ...
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards

Gliederung:

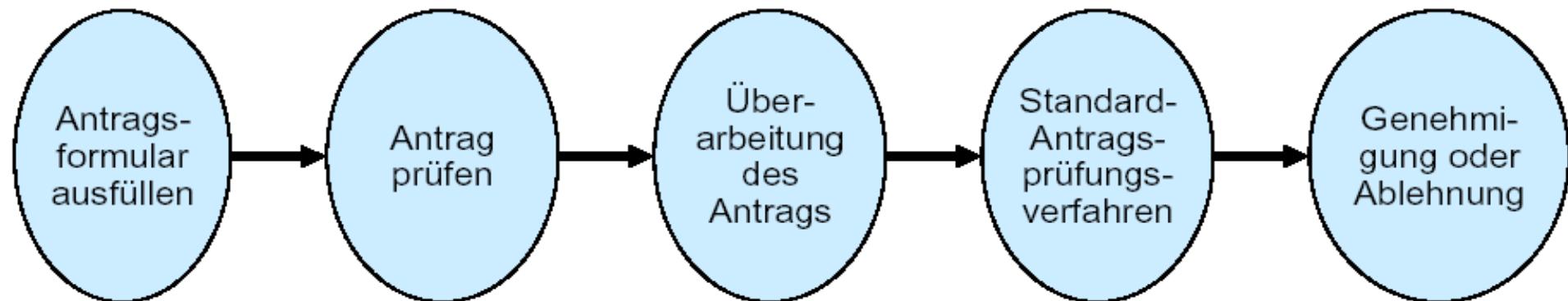
1. Geschäftsprozesse,
2. Geschäftsprozesse und Workflow-Prozesse,
3. Geschäftsprozess- und Workflow-Management,
4. Design von Prozessen,
5. Prozessverbesserung durch BPR,
6. Prozessverbesserung durch CPI,
7. BPR vs. CPI.

1. Geschäftsprozesse

Definition eines Prozesses:

- Ein Prozess ist ein allgemeiner Ablauf mehrerer Abschnitte, bei denen es sich um Aufgaben, Ausführungen, Arbeitsschritte oder Ähnliches handeln kann. Zwischen diesen Prozessabschnitten bestehen bestimmte Abhängigkeiten.

Beispiel: Urlaubsbeantragung



1. Geschäftsprozesse

Definition von Hammer & Champy (1993):

... eine Sammlung von Aktivitäten, die einen oder verschiedene Arten von Input benutzen, um einen Output zu erzeugen, der einen Wert für den Kunden darstellt.

Definition von Davenport (1993):

... eine strukturierte, messbare Menge von Aktivitäten, dafür bestimmt, einen spezifizierten Output für einen bestimmten Kunden oder Markt zu erzeugen. Eine starke Betonung liegt hierbei darauf, **WIE** die Arbeit innerhalb der Organisation ausgeführt wird, im Gegensatz dazu, **WAS** getan wird.

Definition der WfMC (Workflow Management Coalition):

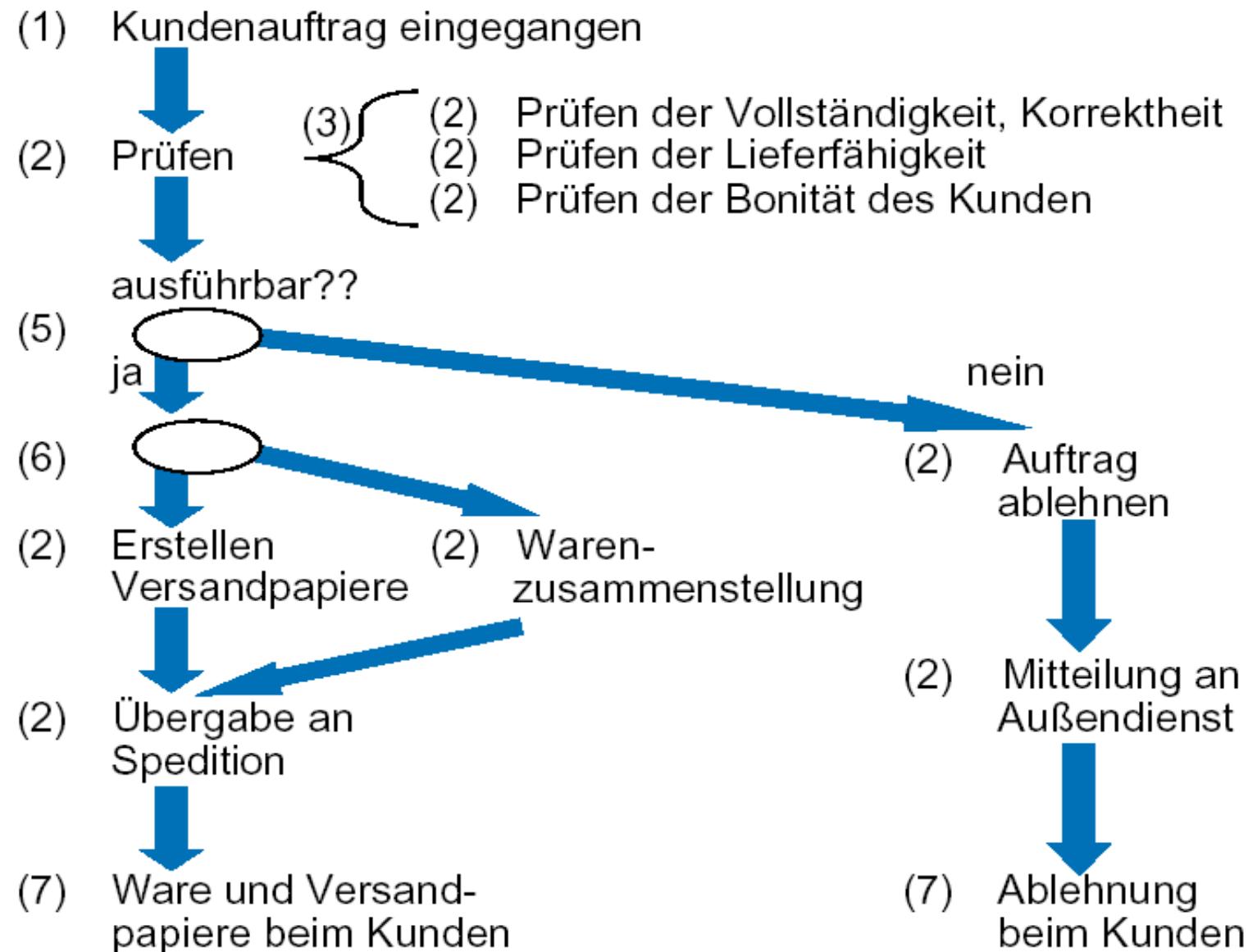
... eine Menge ein oder mehrerer verbundener Arbeitsschritte oder Aktivitäten, die gemeinsam ein Geschäftsziel realisieren oder eine Geschäftsstrategie verfolgen; gewöhnlich wird dies im Kontext einer Organisationsstruktur, die funktionale Rollen und Beziehungen festlegt, betrachtet.

1. Geschäftsprozesse

- Ein **Geschäftsprozess** ist eine Abfolge von Aktivitäten, die der Erzeugung eines Produktes/einer Dienstleistung dienen.
- Er wird durch ein oder mehrere Ereignisse gestartet und durch ein oder mehrere Ereignisse abgeschlossen.
- Es liegt eine Organisationsstruktur zu Grunde.
- Verwendete Synonyme:
 - Ablauf, Vorgang, Prozess, Unternehmensprozess.
- Ein typischer Prozess umfasst:

- { (1) Startereignis (Auslöser)
(2) Aktivität
(3) Zerlegung
(4) Sequenz
(5) Auswahl
(6) Parallelität
(7) Abschlussereignis

1. Geschäftsprozesse, Bsp. Kundenauftrag



1. Geschäftsprozesse / Klassifizierung

Klassifizierung nach Strukturiertheit:

- Strukturierte Vorgänge:
 - Vollständig vorherbestimmt,
 - Wiederholbar,
 - Feste Regelungen für Abwicklung der einzelnen Aufgaben,
 - Einzelaufgaben und ihre Abfolge auf ideale Weise automatisierbar,
- Teil- (Semi-) strukturierte Vorgänge:
 - Enthalten bestimmte Elemente, die sich genau regeln lassen,
 - sowie Elemente, die kaum formalisierbar sind
 - Problemlösungs- oder Entscheidungsfindungsprozessabschnitte,
- Unstrukturierte Vorgänge:
 - Problemlösungssuche / Entscheidungsfindung,
 - Prozesse nicht formalisierbar, verlangen kreativen Freiraum,
 - Dafür geeignet: Workgroup Systeme.

1. Geschäftsprozesse / Klassifizierung

Klassifizierung nach Art des Auftretens:

- Sich zyklisch wiederholende (täglich, wöchentlich), also regelmäßige Geschäftsprozesse mit genau determiniertem Start.
- Vorgänge, die zwar wiederholt auftreten, deren Starttermine aber nicht nach einheitlichen Zeitabschnitten festgelegt werden.
- Einmalige Vorgänge, die sich im Normalfall nicht wiederholen.

Klassifizierung nach Häufigkeit des Auftretens:

- häufig (täglich oder stündlich) auftretende Vorgänge,
- Vorgänge, die je nach Anfragesituation, im Hochbetrieb auch minütlich, abgewickelt werden müssen,
- Vorgänge, die nur manchmal, selten oder auch nur einmal auftreten.

weitere Klassifizierung: externe, interne Vorgänge

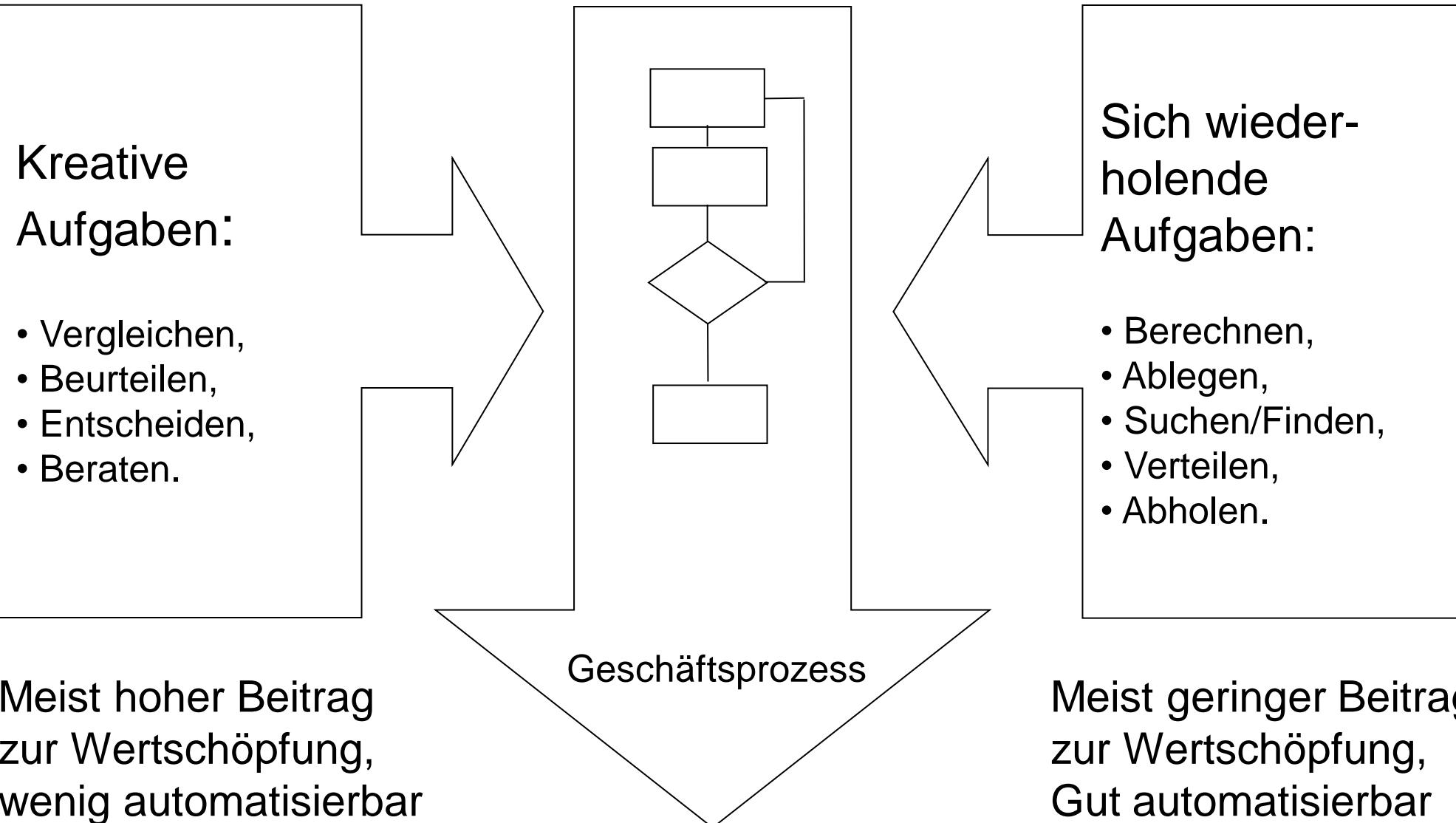
1. Geschäftsprozesse

Einige Entwicklungen, die das stetig wachsende Interesse an optimaler Prozessgestaltung und –verbesserung forcieren:

- ständige Veränderung und Anpassung bei kürzer werdenden Anpassungszyklen,
- zunehmender Kundenfokus,
- höhere Komplexität der Prozesse,
- organisations- und unternehmensübergreifende Prozesse,
- Internet-basierte Prozesse,
- Automatisierung,
 - der Prozessschritte,
 - der Prozessverantwortlichen.

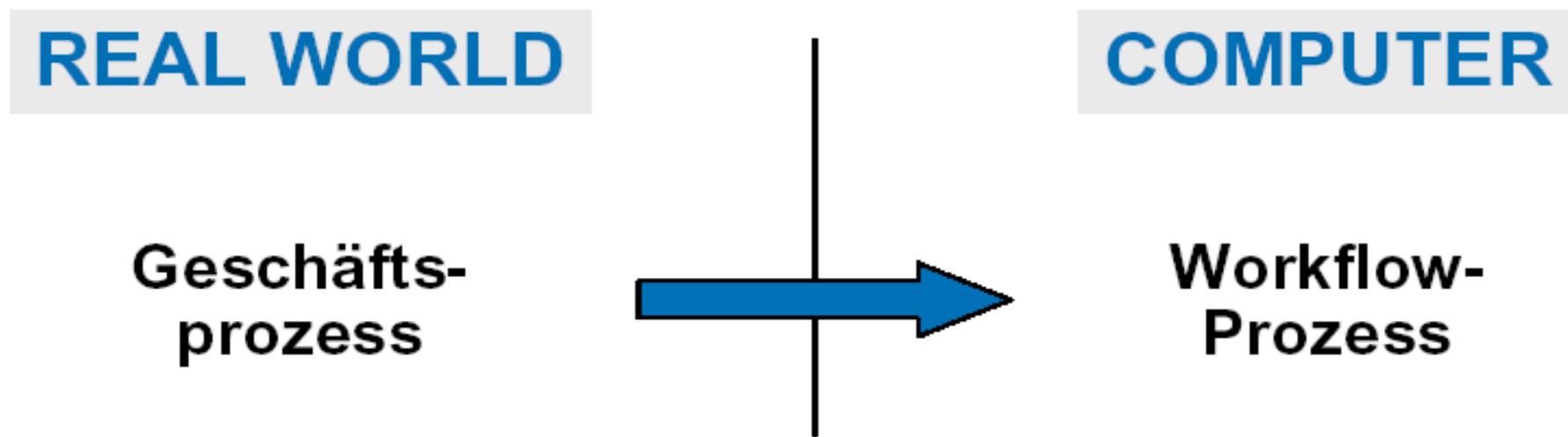
2. Geschäftsprozesse u. Workflow-Prozesse

Automatisierbarkeit von Geschäftsprozessen:



2. Geschäftsprozesse u. Workflow-Prozesse

- Geschäftsprozesse können aus Teilen bestehen, die auf einem Computer ausgeführt werden, sowie aus Teilen, die nicht durch Computer unterstützt werden.
- Definition: Ein **Workflow-Prozess** ist ein zusammenhängender rechnergestützter Teil eines Geschäftsprozesses.



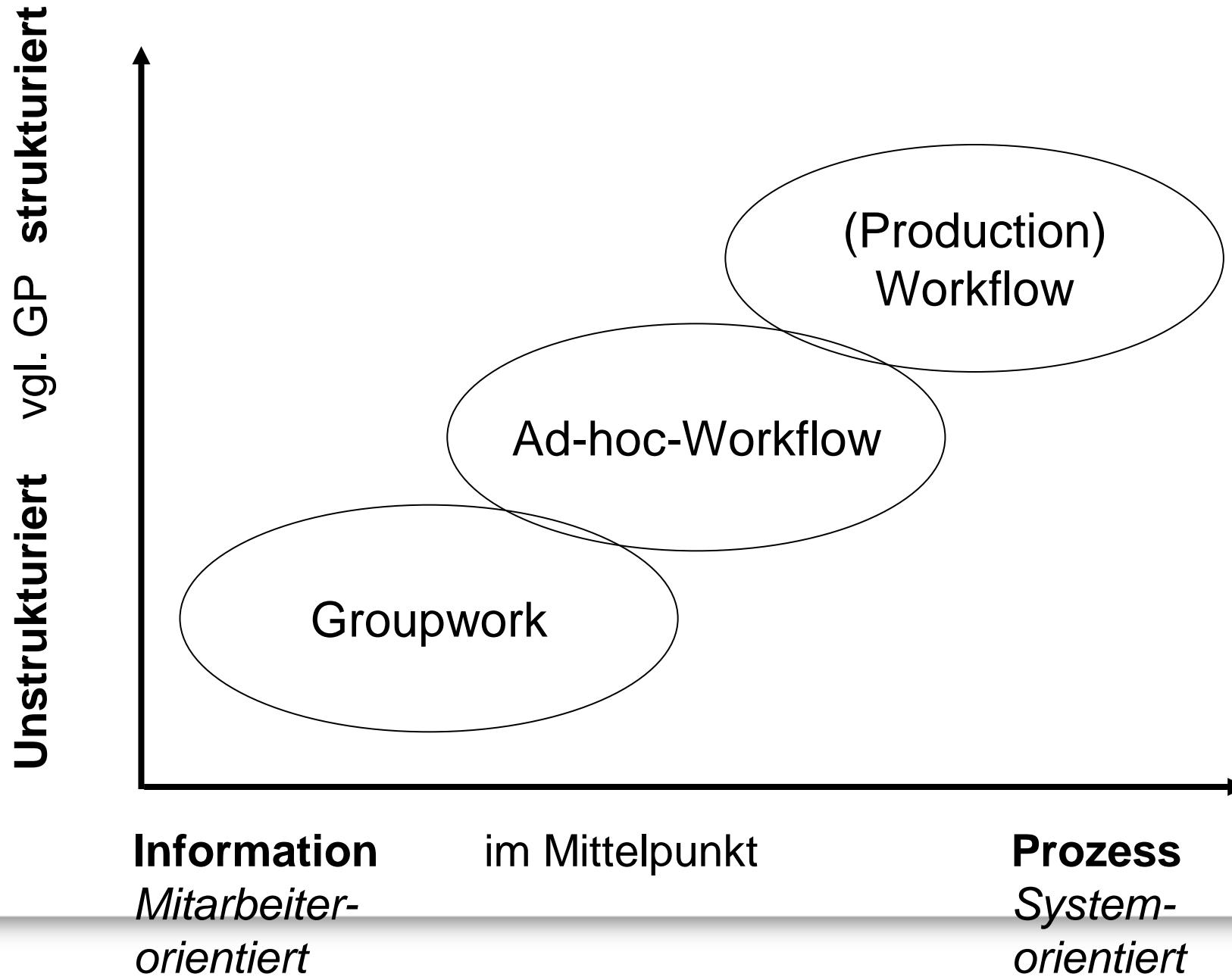
2. Geschäftsprozesse u. Workflow-Prozesse

Workflow-Klassifizierung:

Unstrukturierter Workflow	Semi-strukturierter Workflow	Strukturierter Workflow
Ad-hoc-Entscheidungen	Koexistenz von Ad-hoc-Entscheidungen und strukturierten Elementen	A priori definierte Bearbeitungsstruktur
Nächster Bearbeiter bzw. Bearbeitungsgruppe werden durch Ad-hoc-Entscheidungen festgelegt.	Die Bearbeitungsstruktur ist offen und lässt flexible Ad-hoc-Entscheidungen zu	Nächster Bearbeiter oder Gruppe sind durch vordefinierte Regeln festgelegt
Beispiele: Brainstorming, Literaturrecherche, Problemlösungssuche.	Beispiele: Katalogerstellung, Konstruktion eines neuen Produkts.	Beispiele: Bestellungen, Anträge, Genehmigungsverfahren.

2. Geschäftsprozesse u. Workflow-Prozesse

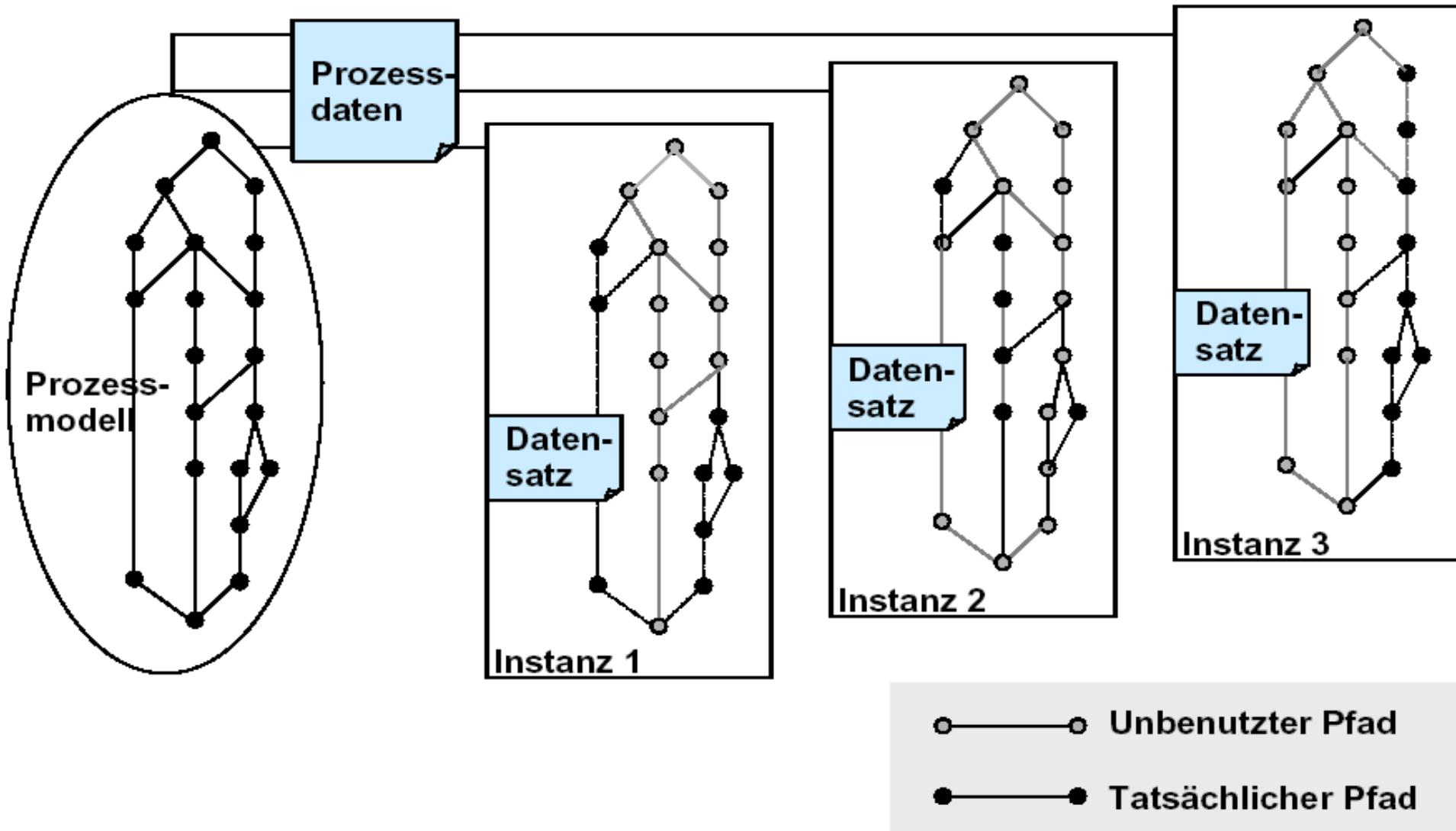
Workflow-Klassifizierung:



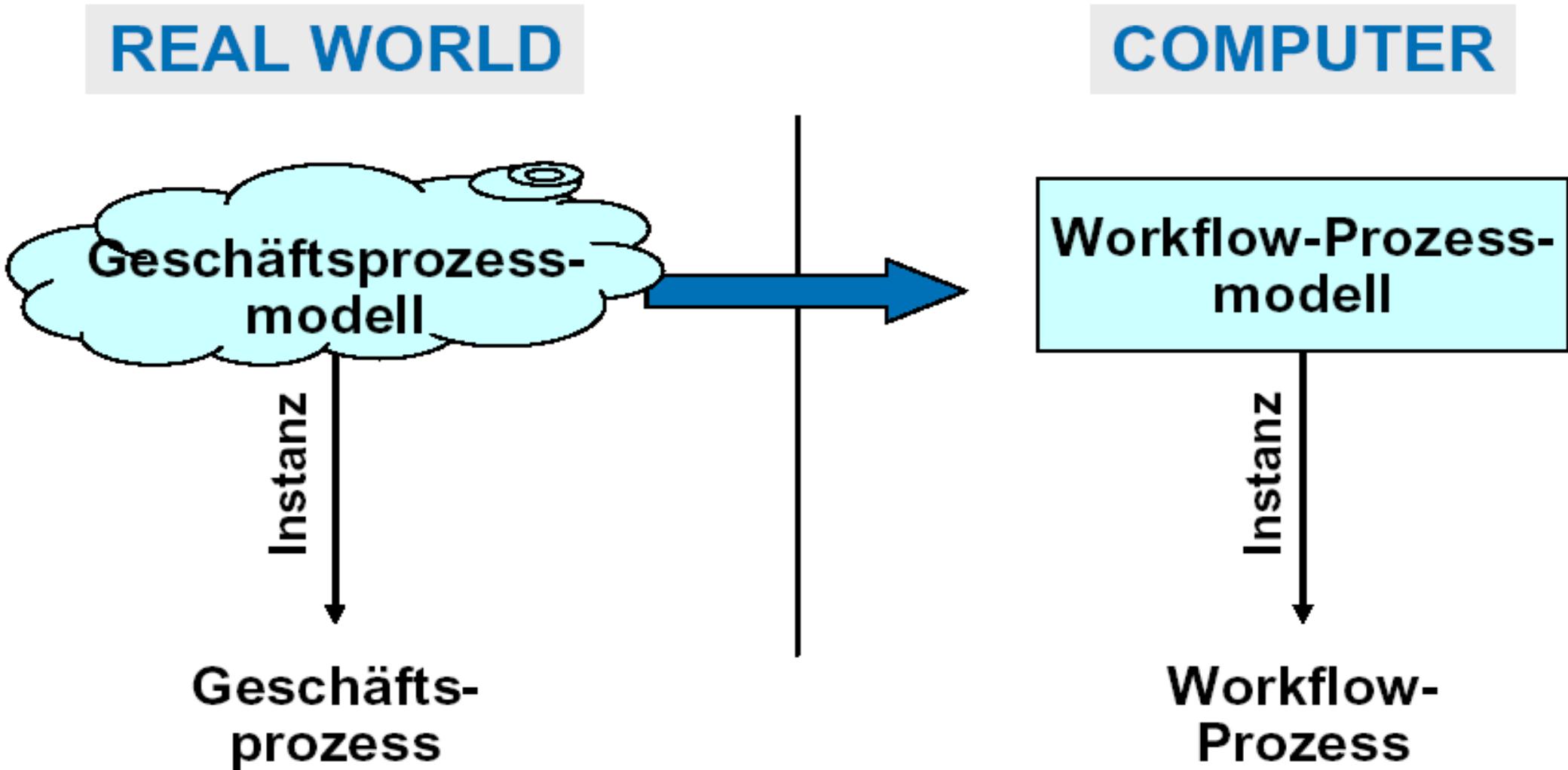
2. Geschäftsprozesse u. Workflow-Prozesse

- Prozessmodell (Prozessdefinition, Prozess-Schema)
 - beschreibt die Struktur eines realen Geschäftsprozesses,
 - bestimmt alle möglichen Pfade/Aktivitäten entlang des Geschäftsprozesses,
 - bestimmt Regeln für die Wahl der Pfade,
 - bestimmt alle Aktivitäten, die ausgeführt werden müssen.
- **Definition:** Ein Prozessmodell ist ein Template (Schablone). Ausgehend davon wird jeder Prozess instanziert.
- **Beispiel:** Versicherung
 - Mögliches Prozessmodell: Schadensbearbeitung
 - Von diesem Modell aus entsteht eine Vielzahl von Prozessen (z.B. für jeden Klienten, nach Art des Fahrzeugs, Art des Schadens, ...).

2. Geschäftsprozesse u. Workflow-Prozesse



2. Geschäftsprozesse u. Workflow-Prozesse



- **Ziel:**

Arbeitsfluss so organisieren, dass die richtige Arbeit zum richtigen Zeitpunkt von der richtigen Ressource (Person) ausgeführt wird.
(automatisierte Prozesssteuerung).

- **Lösung eines „alten“ Problems**

Controlling, Monitoring, Verbesserung und Unterstützung von Geschäftsprozessen.

- **Neuer Aspekt**

Die explizite Darstellung der Logik von Geschäftsprozessen erlaubt die Unterstützung durch IT.

■ Ziele und Vorteile:

- Kontrolle, Verbesserung der Prozessabwicklung / Verteilte Prozessabwicklung,
- Transparenz der Arbeitssituation / Verbesserte Arbeitsvorratsverwaltung,
- Koordination räumlich oder zeitlich verteilter „Bearbeiter“,
- Flexibilitätssteigerung,
- Verkürzung der Durchlaufzeiten,
- Gemeinsame Nutzung von Dokumenten,
- Einheitliche Benutzeroberfläche,
- Qualitätssicherung,
- Besserer Kundenservice,
- Investitionsschutz,
- Rationalisierung (mit oder ohne BPR),
- Weiterführung der ISO9000-Arbeiten,
- Produktivitätssteigerungen.

- **Nachteil:**

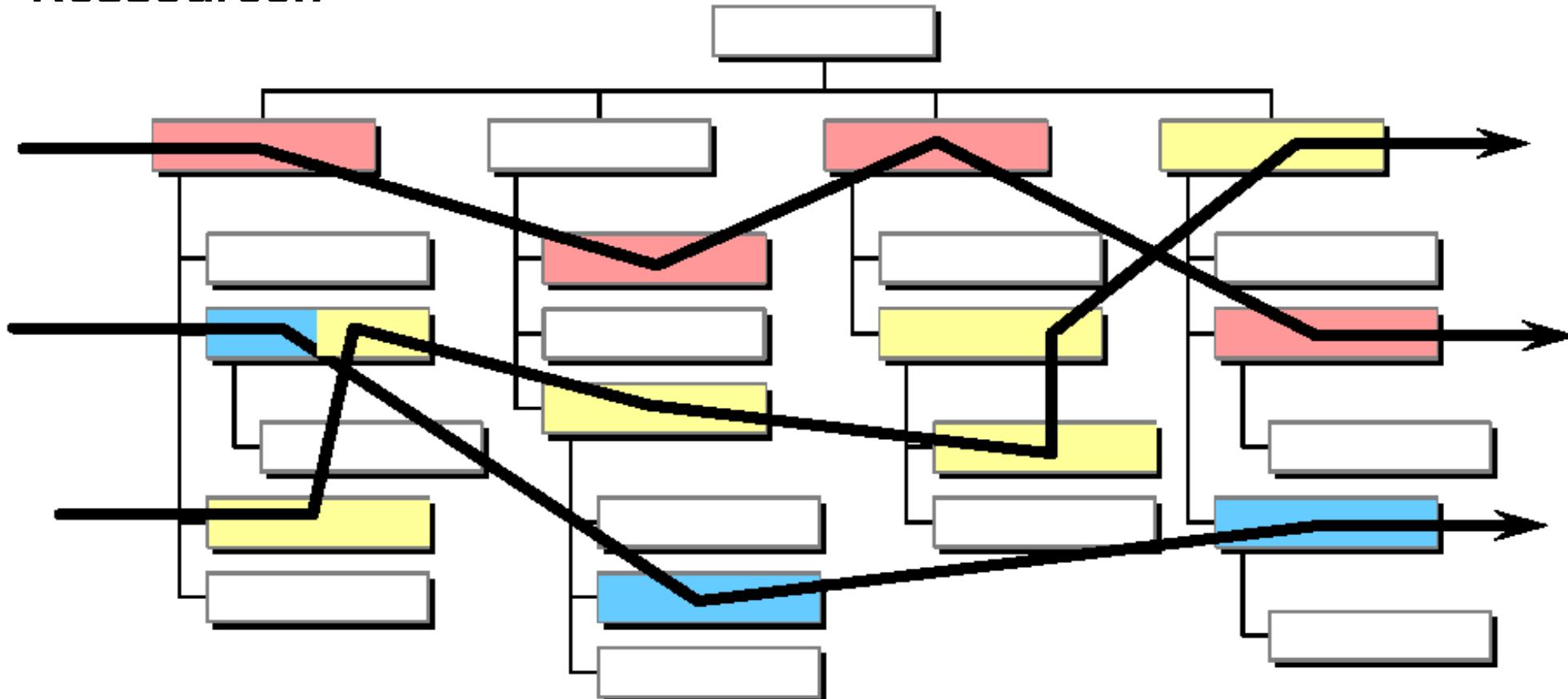
- Mängel im Sicherheitsbereich (Datenschutz)

- **Einsatzbereiche:**

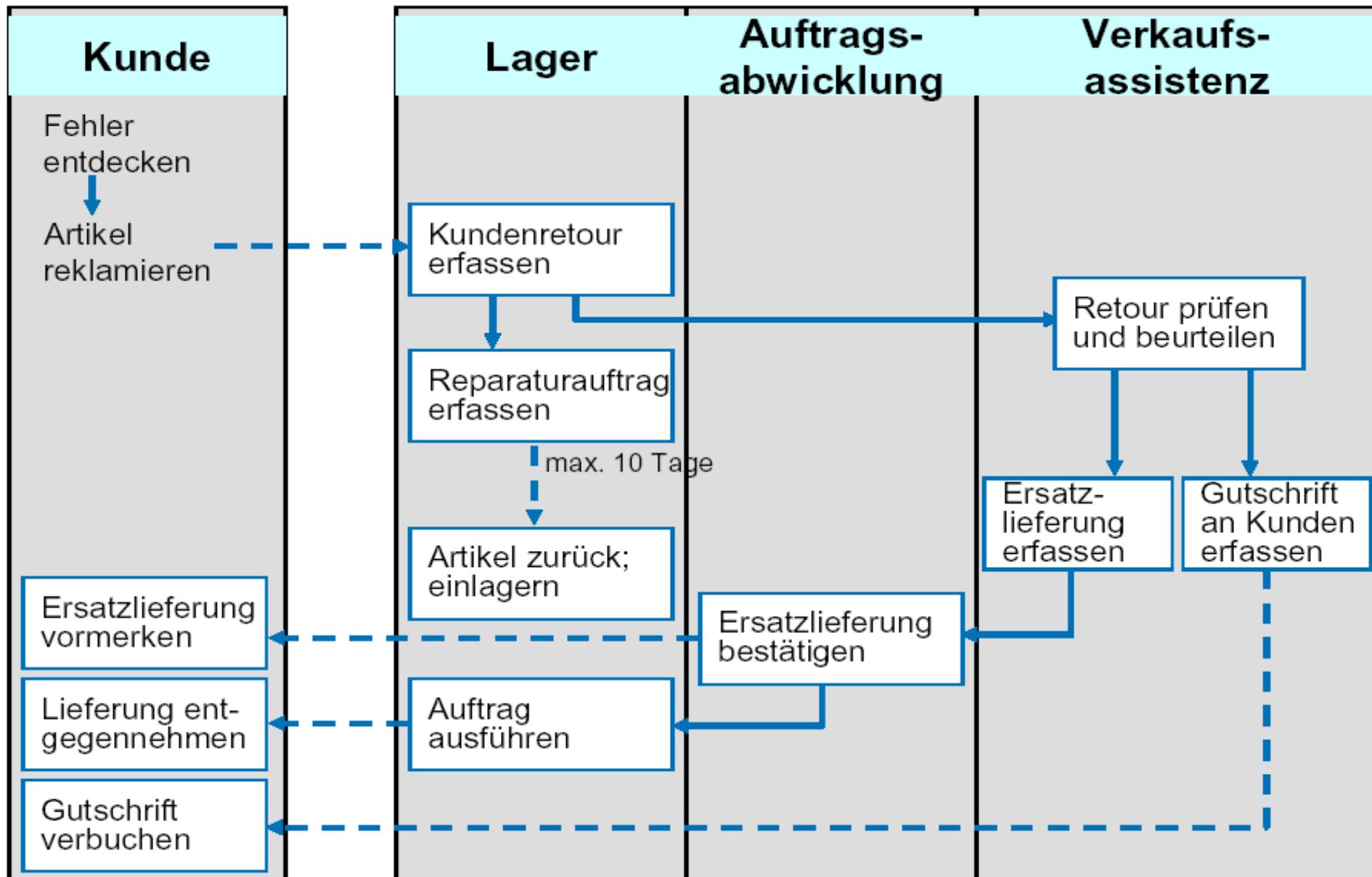
- Abläufe, in die eine hohe Anzahl von Personen oder Applikationen einbezogen sind (Bsp.: Schadensabwicklung bei einer Versicherung),
 - Abläufe mit hohem Strukturierungsgrad und geringer Komplexität (Bsp.: Auftragsbearbeitung in einem Handelsbetrieb),
 - Abläufe mit hoher Wiederholungsfrequenz, die wenige Ausnahmebehandlungsmechanismen erfordern (Bsp.: Zahlungsverkehr in einer Bank)
 - Strategisches Management.

Geschäftsprozesse und Organisationstheorie:

- entscheidende Rolle spielt die Aufbauorganisation und damit die Ressourcen



■ Beispiel: Retourenabwicklung



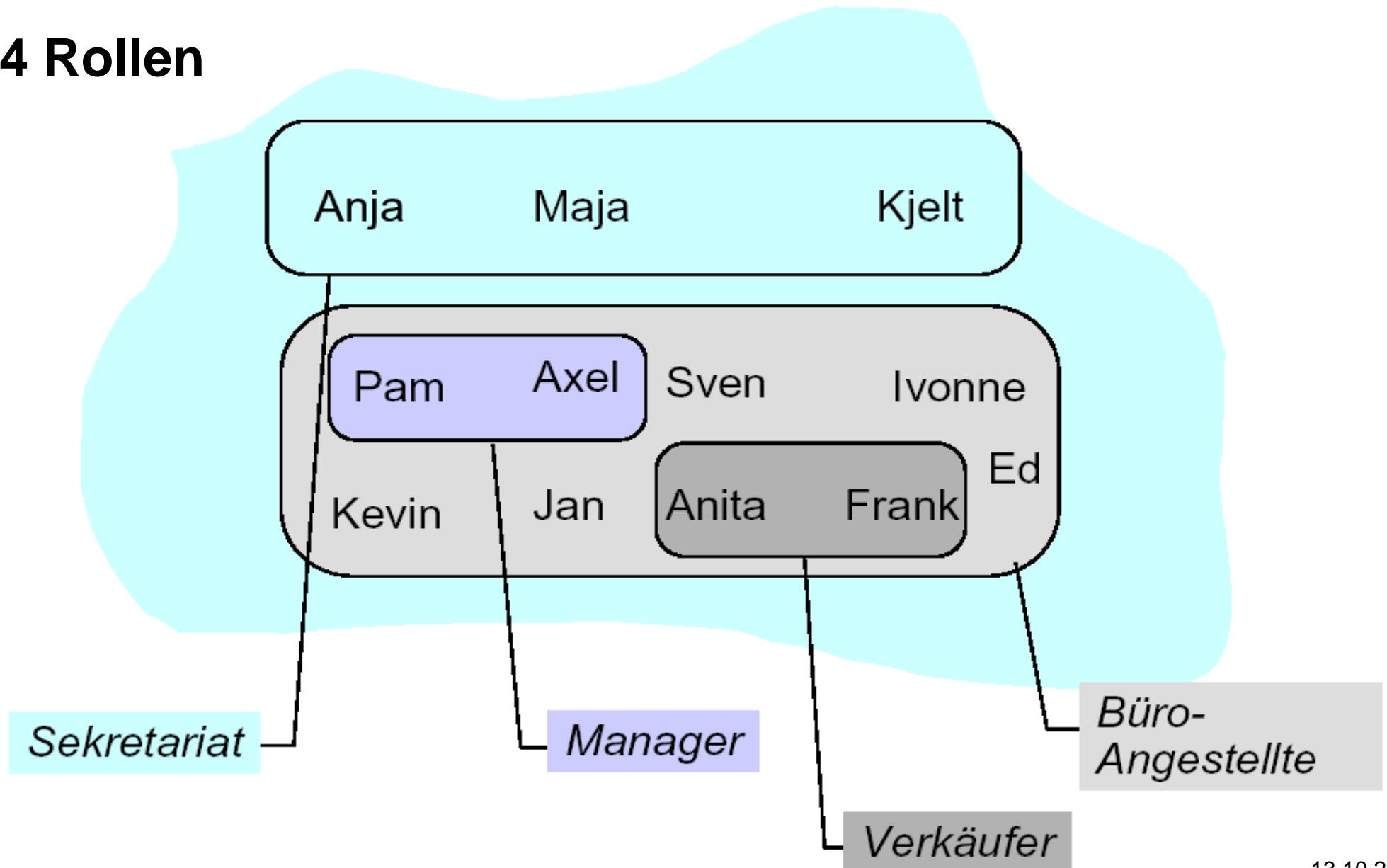
Ressourcenklassifikation:

- Eine **Ressourcenklasse** ist eine Menge von Ressourcen mit ähnlichen Eigenschaften.
- **Organisationseinheiten** sind Ressourcenklassen, die sich aus der Struktur der Organisation ableiten (auch: Gruppe, Abteilung, Team, ...). Können aus einem Organigramm abgelesen werden.
- **Rollen** sind Ressourcenklassen, die sich aus den Fähigkeiten der Ressourcen ableiten (auch: Skills, Kompetenzen, Befugnisse, ...). Sind meist nicht direkt aus einem Organigramm ablesbar.

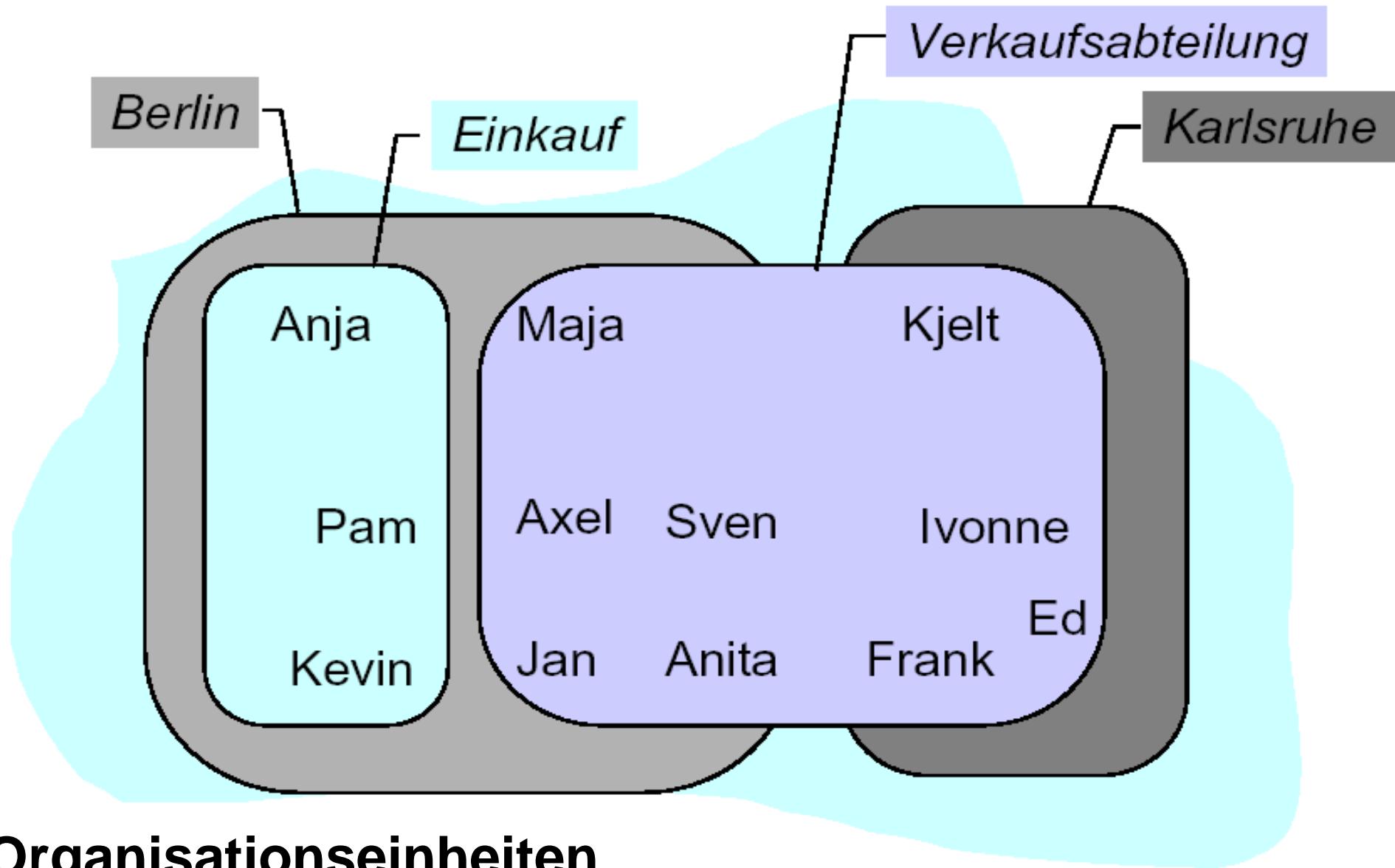
12 Ressourcen



4 Rollen

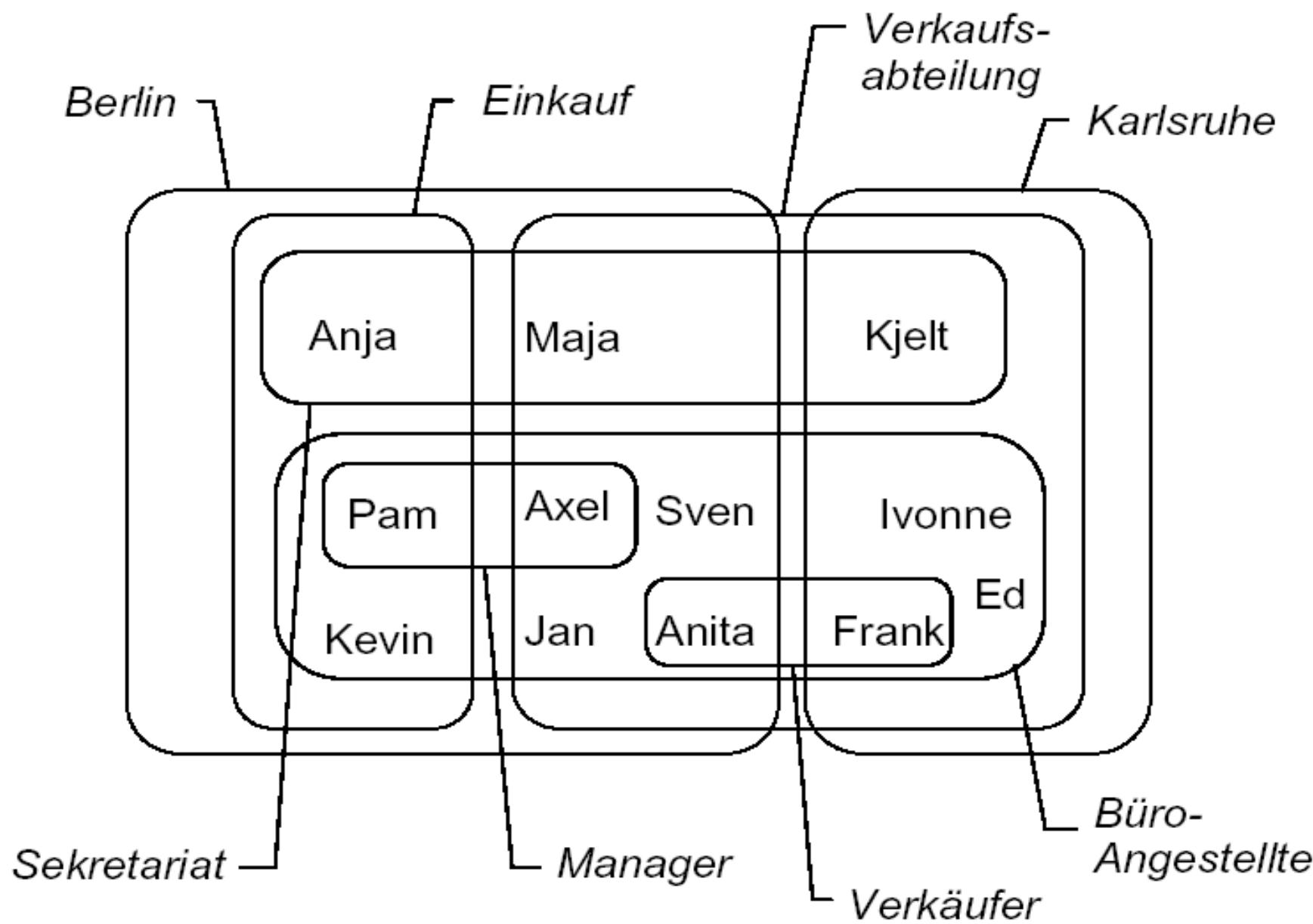


3. Geschäftsprozess- und Workflow-Management



4 Organisationseinheiten

3. Geschäftsprozess- und Workflow-Management



8 Ressourcenklassen



Beispiel: IT-Unternehmen

Abt...

Abt. Forschung

Abt. Ausbildung

Abt...

Rosi Schmidt

Lars Kühn

Sekretariat

Sven Süss

Pam Jones

Wirtschaftsinformatik

Hardware

Anja Klein

Software

Workflow nach v.d. Aalst::

- Ein **Fall** ist Auslöser einer Instanz / eines Prozesses.
- Ein Fall ist etwas, das entsprechend der Prozessdefinition bearbeitet (abgewickelt, behandelt o.ä.) werden muss.
- Eine entsprechende Prozessdefinition legt die **Aufgaben und ihre Reihenfolge** fest.
- Aufgaben werden für einen speziellen Fall ausgeführt.
- Fälle werden unabhängig voneinander behandelt.

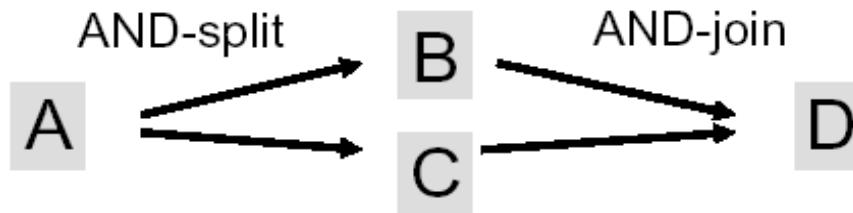
Workflow nach v.d. Aalst:

- **Fälle** werden bearbeitet, indem **Aufgaben** (hier A, B, C, D) in einer bestimmten Reihenfolge ausgeführt werden. (Routing (Weiterleitung) von Fällen)

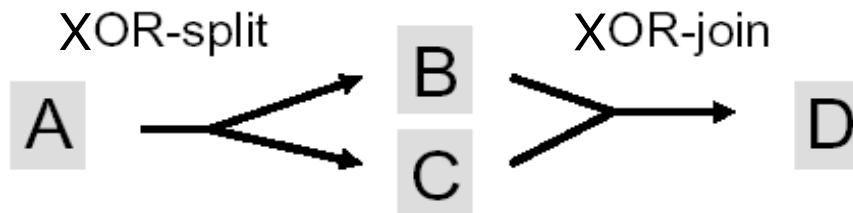
- sequentiell



- parallel



- alternativ



- iterativ



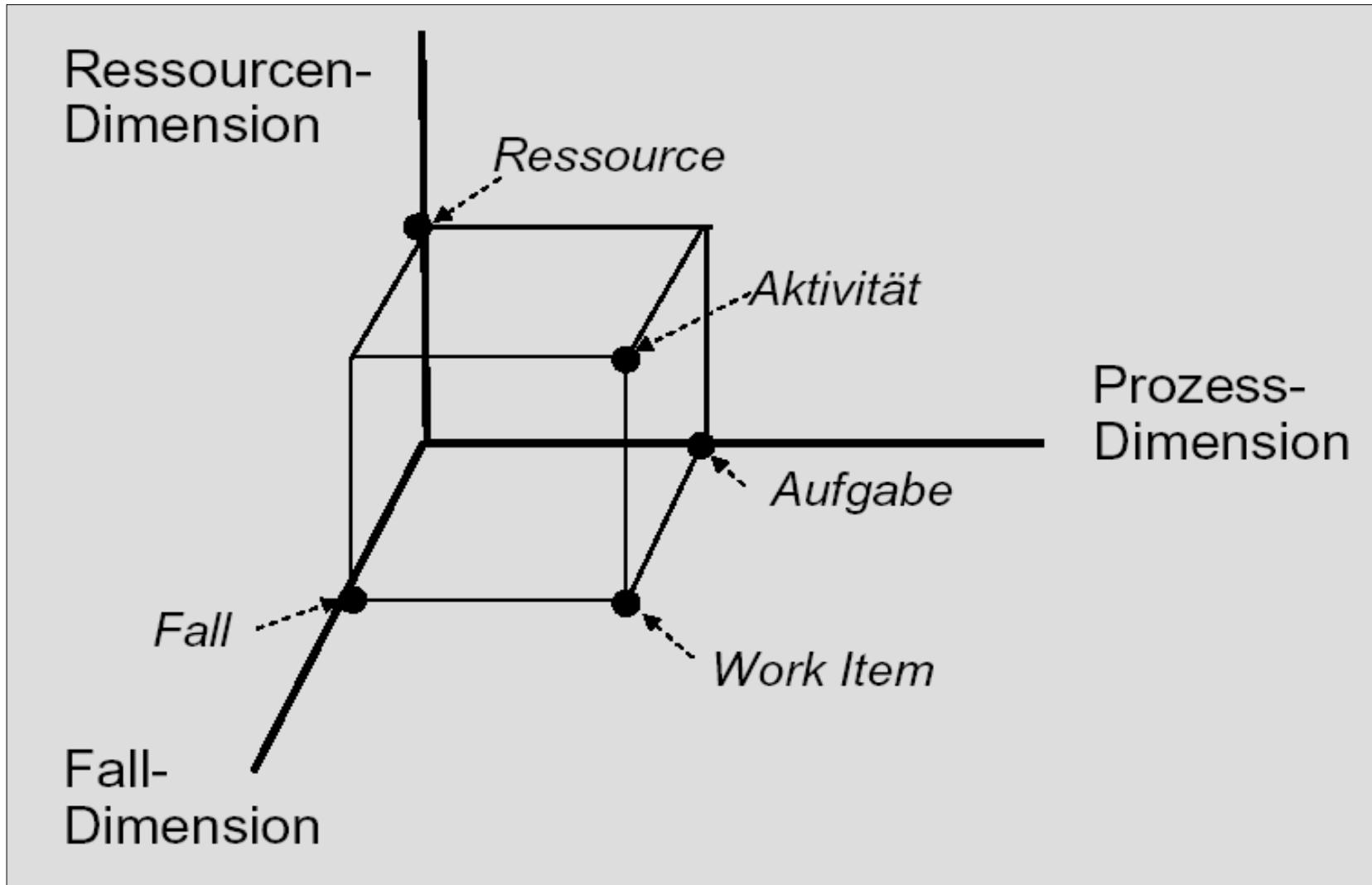
Workflow nach v.d. Aalst::

- Ein **Workflow-Prozess** dient der Bearbeitung eines Falles.
- Bedingungen legen die Reihenfolge der Aufgaben fest, sie können wahr oder falsch sein.
- Eine Aufgabe hat Vorbedingungen (Anforderungen die für die Weiterleitung eines Falles, erfüllt sein müssen) und Nachbedingungen.
- Fälle sollten so effektiv und effizient wie möglich behandelt werden: maximaler **Kundennutzen**.
- Beispiele: Versicherungsfall, Kauforder, Beschwerde, ..

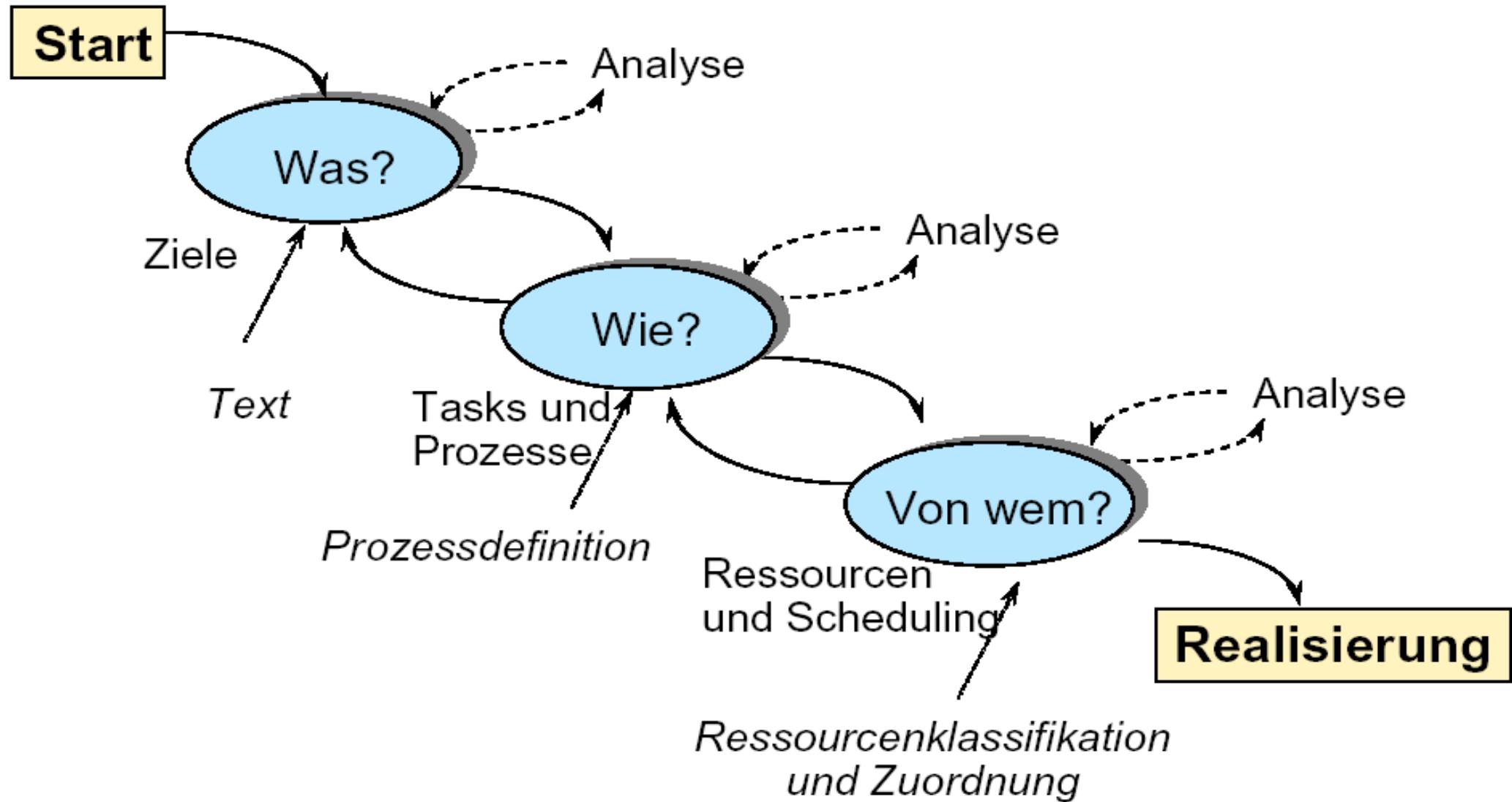
Workflow nach v.d. Aalst::

- **Aufgabe (task):** Schritt innerhalb der Prozessdefinition,
- **Work Item:** Aufgabe, die für einen bestimmten Fall durchzuführen ist
Work Item = (Aufgabe, Fall),
- Work Items werden von Ressourcen ausgeführt. Dies kann eine Maschine, Software oder eine Person sein.
- **Aktivität:** Work Item, das von einer bestimmten Ressource (oder mehreren) bearbeitet wird.
Aktivität = (Aufgabe, Fall, Ressource).

Workflow nach v.d. Aalst: 3-dimensionale Ansicht eines Workflows



4. Design von Prozessen



Beginn mit der **Identifizierung des Falles** (Case),

- Ein Fall wird meistens durch einen Kunden (intern oder extern) initiiert.
- Der Prozess sollte zu einer Wertschöpfung bezüglich des Falles führen.
- Ein Fall besitzt einen Lebenszyklus mit Beginn und Ende.
- Ein Fall kann nicht unterteilt werden, die Arbeit jedoch schon.

Definition des Prozesses kann aus der Fallidentifizierung abgeleitet werden:

- möglichst genaue Festlegung des Prozess-**Ziels**,
- möglichst frühe Festlegung des Prozess-**Umfangs**,
- Festlegung der logischen **Abhängigkeiten**,
- Abstraktion von Ressourcen beim Entwurf des Prozess-Modells,
- Prozess-Entwurf ist **iterativer** Entwicklungsvorgang (keine Angst vor Fehlern!),
- Granularität der Aufgaben/Tasks verändert sich,
- Hierarchie-Konzepte sollten genutzt werden (**top-down**).

Aufgaben als LUWs (Logical Units of Work) während des Design-Prozesses

- eine Aufgabe wird von einer Ressource zu einer Zeit an einem Ort ausgeführt,
- atomar: *Commit* (vollständige Ausführung und Übergabe) oder *Rollback* (rückgängig, falls Commit nicht möglich),
- Rüstzeiten sollten minimiert werden,
- Umfang der LUW's sollte so klein wie möglich und nur so groß wie notwendig werden (Commit zum frühestmöglichen Zeitpunkt).

Ressourcenaufstellung parallel zum Entwurfsprozess

- Prozessentwurf zunächst **ohne** Ressourcenberücksichtigung,
- Festlegung des Umfangs des Aufgaben **mit** Ressourcenbezug!

ACID-Eigenschaften: aus der Datenbank-Welt, Transaktionsverarbeitung

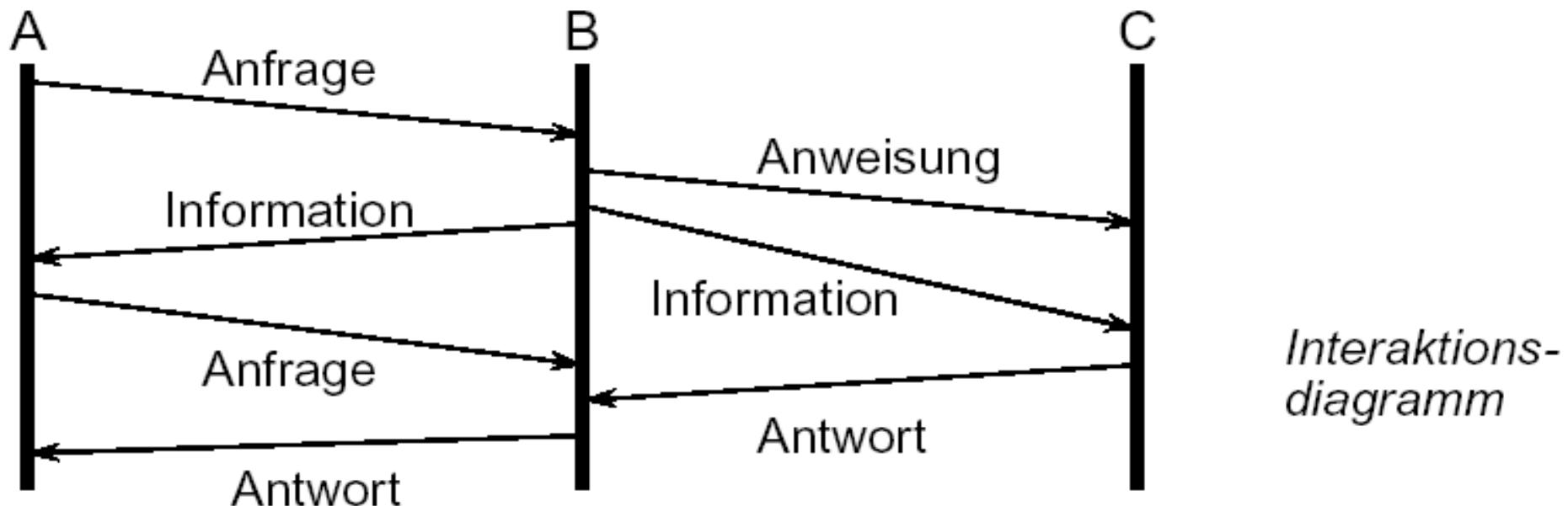
- Atomicity: ("alles oder nichts", Commit oder Rollback),
- Consistency: (eine beendete Task überführt das System in einen gültigen Zustand),
- Isolation: (Tasks beeinflussen sich nicht gegenseitig, auch wenn sie parallel ausgeführt werden),
- Durability: (Ergebnisse einer komplettierten Task gehen nicht verloren).

4. Design von Prozessen

Informationen über einen Prozess sammeln:

- Wege der Dokumente bestimmen,
- Auswertung der Kommunikationswege zwischen Einzelpersonen, Teams, Abteilungen,

Kommunikationswege feststellen durch Interaktionsdiagramme:

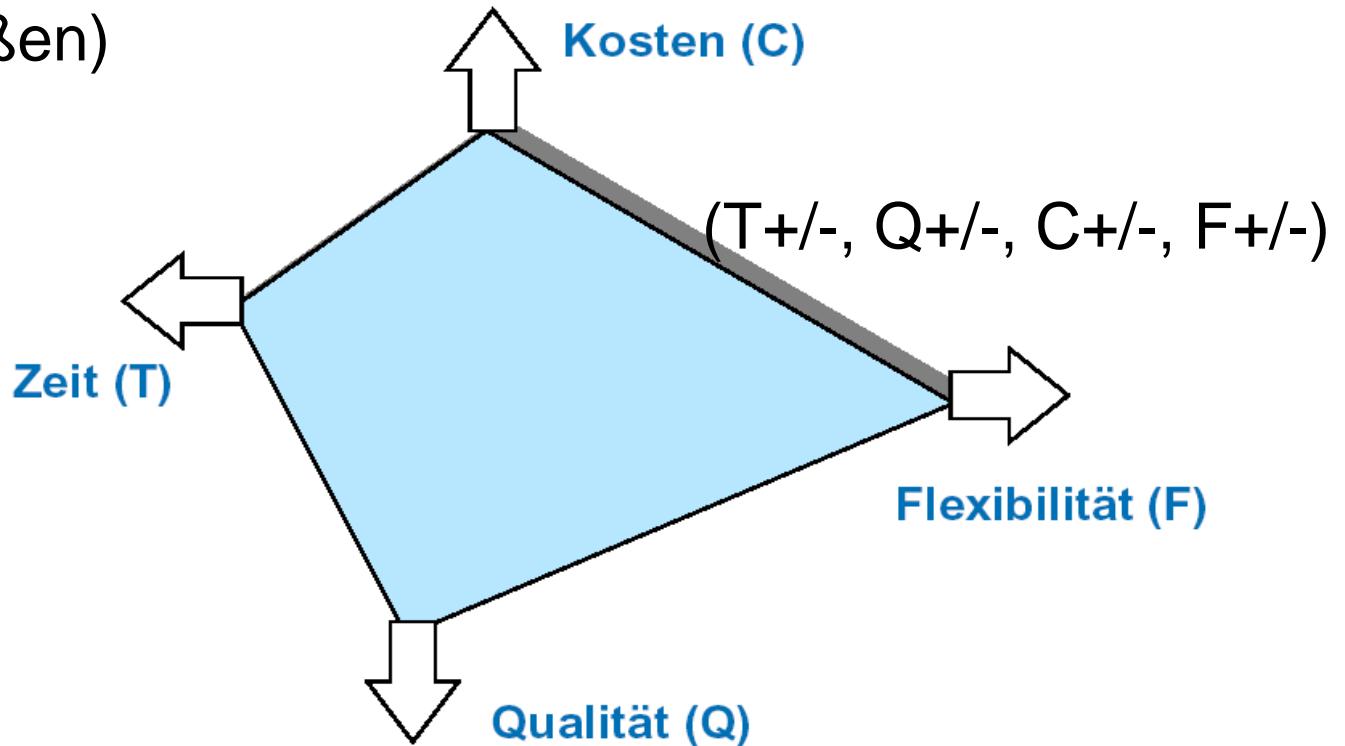


4. Design von Prozessen

Vier Kriterien zur Bewertung eines Prozess-Designs:

(= kritische Leistungsgrößen)

- Zeit (T),
- Qualität (Q),
- Kosten (C),
- Flexibilität (F).



Problem: in der Regel kommt es zu einem Trade-Off (Abwägen der Kriterien).

4. Design von Prozessen

Zeit oft als **Durchlaufzeit**, die sich zusammensetzt aus aus

- Servicezeit (inklusive Rüstzeit),
- Transportzeit (kann oft verhindert werden),
- Wartezeit,
 - limitierte Ressourcen,
 - externe Einflüsse (Trigger).

Kennzahlen im Zusammenhang mit Durchlaufzeit

- Durchschnitt,
- Varianz,
- Service-Level,
- Termintreue.

4. Design von Prozessen

Externe Qualität:

- Zufriedenheit des Kunden in Bezug auf
 - Produkt: entspricht den Anforderungen/Erwartungen,
 - Prozess: Service-Level,

Interne Qualität:

- Arbeitsbedingungen, z.B.
 - Anspruch,
 - Abwechslung,
 - Kontrolle, ...

Anmerkung: Es existiert oft eine positive Korrelation zwischen interner und externer Qualität.

4. Design von Prozessen

Kostenarten:

- fixe und variable Kosten,
- Arbeits-, Personal- und Systemkosten ,
- externe Kosten,
- Bearbeitungs-, Verwaltungs- und Support-Kosten,
- ...

Anmerkung: Es kommt oft zu einem Trade-Off zwischen Arbeits-, Personal- und Systemkosten.

WICHTIG: Fähigkeit, auf Veränderungen zu reagieren

Flexibilität in Bezug

- **Ressourcen:** Fähigkeit, verschiedene und neue Tasks auszuführen,
- **Prozess:** Fähigkeit, verschiedene Fälle handhaben zu können, verschiedene Auslastungen zu verkraften,
- **Management:** Fähigkeit, Regeln und Ressourcen-Allokation zu ändern,
- **Organisation:** Fähigkeit, die Struktur den Anforderungen des Marktes und der Partner anzupassen.

4. Design von Prozessen / Möglichkeiten (1)

- Prozess- und Case-Verantwortlichen bestimmen
- Notwendigkeit von Tasks überprüfen,
- Größe der Tasks überdenken,
- Trade-Off zwischen einem generischen Prozess und verschiedenen Varianten für einen Prozess überdenken,
- Trade-Off zwischen generischen Tasks und verschiedenen spezialisierten Tasks überdenken,
- Grad der Parallelität erhöhen (falls möglich).

4. Design von Prozessen / Möglichkeiten (2)

- neue Möglichkeiten durch Einsatz von Technologie untersuchen
- Kommunikationswege überprüfen und optimieren,
- Bedenken: ein elektronisches Dokument ist überall und nirgends,
- Ressourcen dazu nutzen, wozu sie gut sind,
- Möglichst hohe Flexibilität bewahren,
- Rüstzeiten (Setup Time) verringern,
- Setups verringern,
- **Nicht den Weg des Papiers automatisieren!**

Prozess- und Case-Verantwortlichen bestimmen:

Ein Prozess-Verantwortlicher überwacht den Prozess und achtet auf Engpässe, Kapazitätsprobleme und Fristen.

Case-Verantwortliche werden einem Fall (Case) zugeordnet. Sie sind für die korrekte Bearbeitung des Falls zuständig.

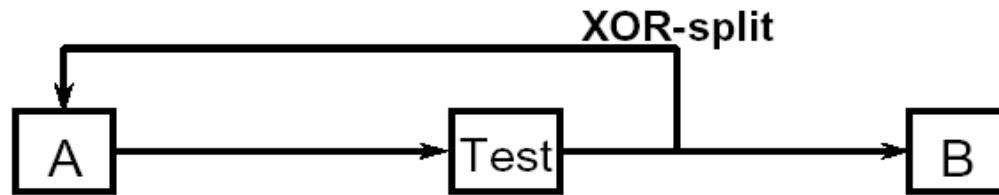
Vorteile:

- Verantwortung (Commitment),
- 1 Ansprechpartner,
- Verringerung der Rüstzeit (Setup Time) bei Cases.

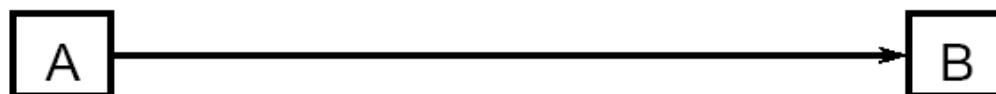
(Q+)

4. Design von Prozessen

- Notwendigkeit von Tasks überprüfen

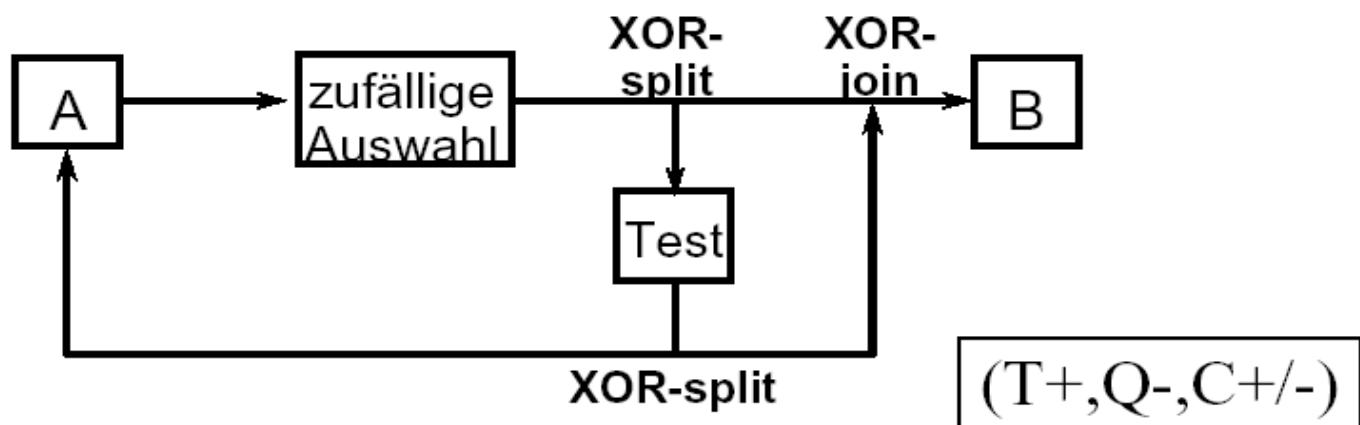


- jede Test-Aufgabe kann ausgelassen werden:



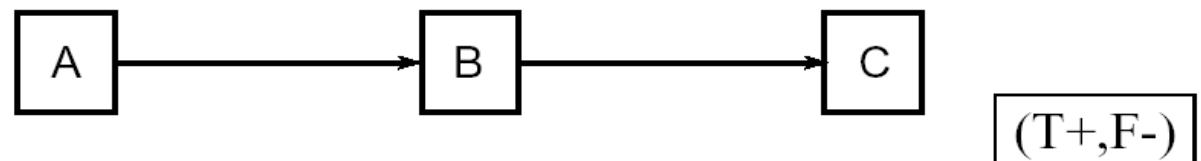
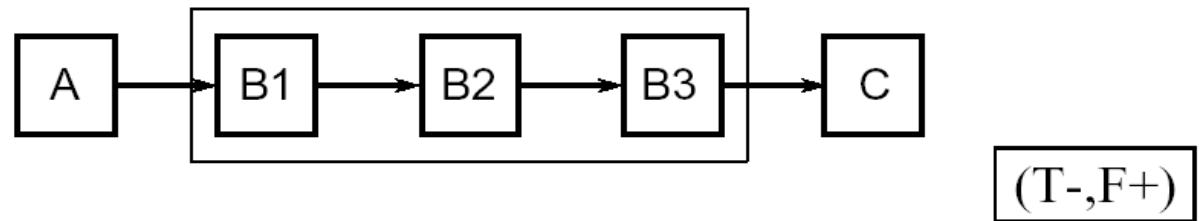
- **ACHTUNG:** Trade-Off zwischen den Kosten der Überprüfung und den Kosten durch Nicht-Überprüfung!

- Alternative:



Größe der Tasks überdenken:

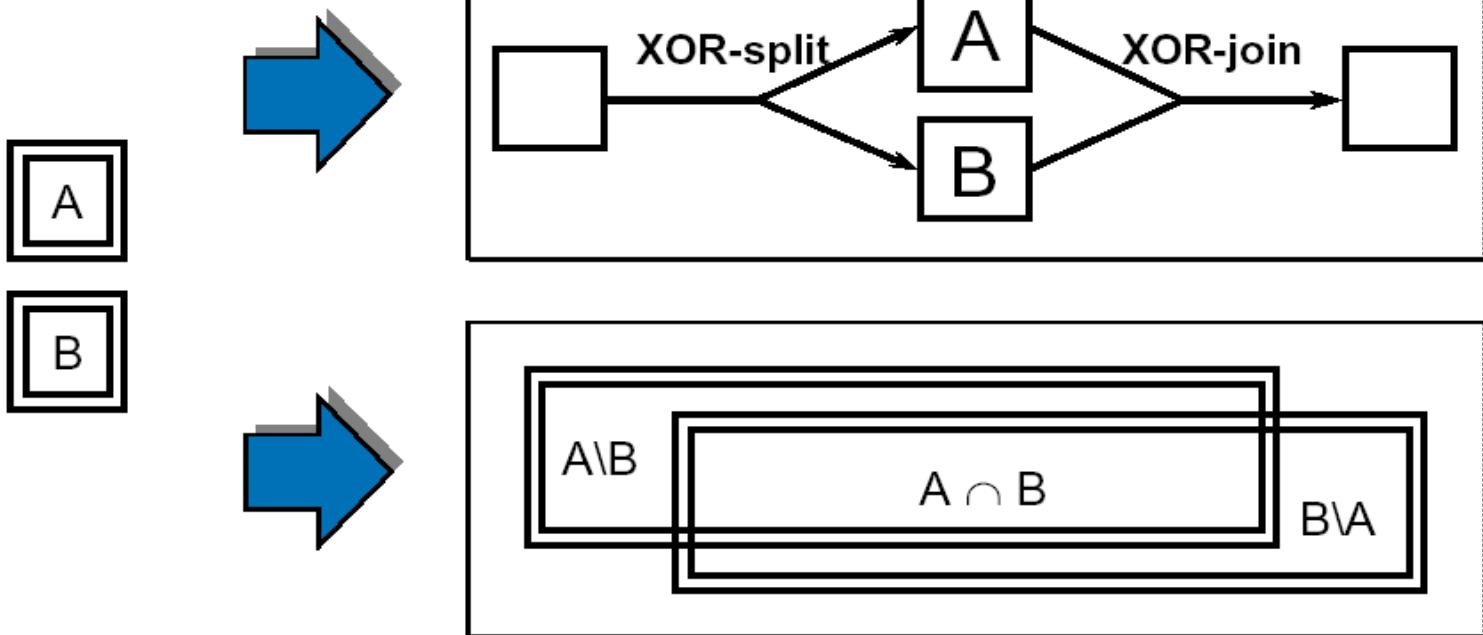
- *Für* : weniger Arbeit pro Aufgabe, ermöglicht Spezialisierung,
- *Wider* : Rüstzeit, Fragmentierung, weniger Verantwortung



- *Für* : weniger Rüstzeit, ...,
- *Wider* : erfordert Qualifizierung bzgl. der einen Aufgabe

Anmerkung: Trade-Off zwischen Komplexität des Prozesses und Komplexität der Tasks!

Generische Prozesse vs. Varianten:



- *Punkte:* Einfachheit, Effizienz, Kontrollierbarkeit, Wartbarkeit,...

(F+/-)

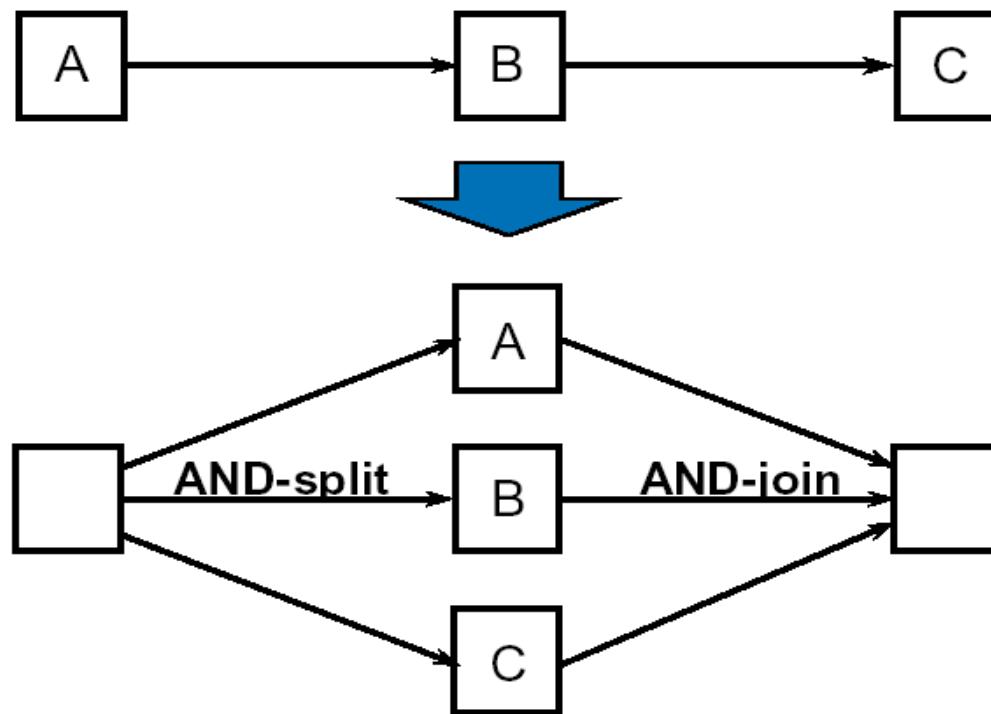
Generische Tasks vs. Varianten:

- vgl. generische Prozesse
- Spezialisierung kann die folgenden Auswirkungen haben:
 - verbesserte Allokation der Ressourcen,
 - bessere Ausführung der Tasks,
 - weniger Flexibilität,
 - komplexere Prozesse,
 - Monotonie.

(T+, F-)

Parallelität:

- Hoher Grad an Parallelität führt zu einer besseren Performance:
- (Verringerung der Wartezeiten und bessere Nutzung von Kapazitäten)
- benötigt IT-Infrastruktur, die gleichzeitige Nutzung von Daten ermöglicht!



(T++)

Einsatzmöglichkeiten für IT:

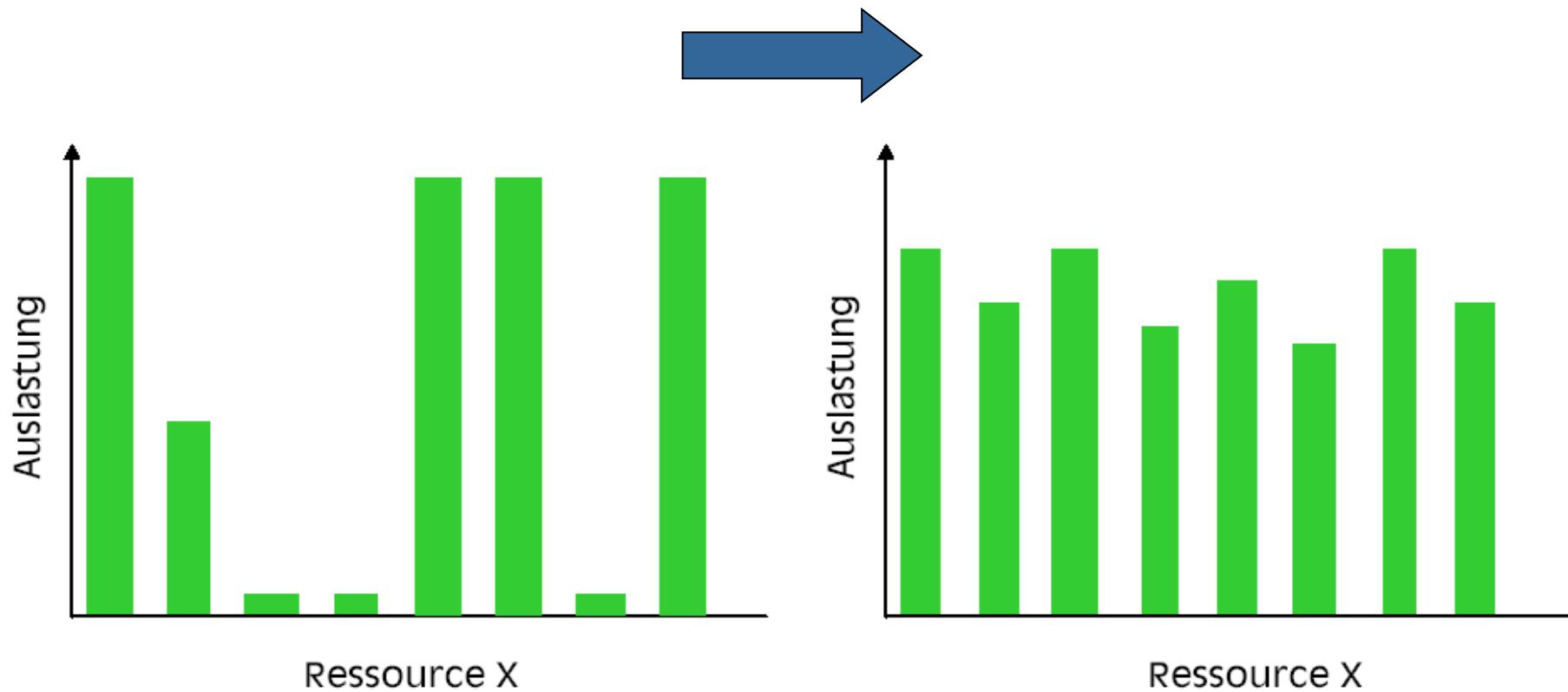
- DBMS: Sharing von Daten → ermöglicht Parallelität,
- Netzwerke
 - Kommunikation: e-mail, WWW, ...
 - → kann den Kundenbezug verbessern,
 - → ermöglicht asynchrone Kommunikation.
 - Verteilung von Information
 - Transport von Daten ist schnell und kostengünstig
- Automatisierung von Aufgaben oder automatisierter Support bei der Ausführung (bspw. Analyse historischer Daten, Risikoanalyse, ...)

(T+, Q+/-, C+/-, F-)

4. Design von Prozessen

Ressourcen gleichmäßig auslasten:

- Vermeiden von Überlastung, während andere warten



Ressourcen ihren Fähigkeiten entsprechend nutzen:

- ... jedoch auf Inflexibilität durch Spezialisierung achten!
- Routine entstehen lassen,

Rüstzeit so gut wie möglich verringern:

- für Cases,
- für Tasks.

(T+, Q+/-)

Kommunikationswege überprüfen:

- Anzahl der auszutauschenden Nachrichten verringern (insbesondere externe Kommunikation),
- Handhabung der Nachrichten automatisieren,
- Kommunikationsfehler vermeiden (EDI, WWW),
- wenn möglich asynchrone statt synchroner Kommunikation verwenden (Email <> Telefon)

(T+, Q+, C+/-, F-)

Geschäftsprozess-Management

Prof. Dr.-Ing. Andreas Ittner

Email: ittner@hs-mittweida.de

WWW: www.hs-mittweida.de/~ittner

Tel.: +49(0)3727-58-1288

Mob.: +49(0)177-5555-347

Gliederung (vorläufig)

- Motivation
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - **Prozessdesign, Prozessverbesserungen**
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs, BPMN, ...
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards

Gliederung:

1. Geschäftsprozesse,
2. Geschäftsprozesse und Workflow-Prozesse,
3. Geschäftsprozess- und Workflow-Management,
4. Design von Prozessen,
5. Prozessverbesserung durch BPR,
6. Prozessverbesserung durch CPI,
7. BPR vs. CPI.

5. Prozessverbesserung durch BPR

- BPR = Business Process Reengineering,
- 3 Faktoren:
 - die Kunden übernehmen das Kommando,
 - der Wettbewerb wird intensiver,
 - der permanente Wandel wird zur Konstante.
- Hammer und Champy: „Reengineering the Corporation“ (1993): “The fundamental rethinking and radical redesign of business processes to bring about dramatic improvements in … cost, quality, service and speed.”
- Existierende Prozesse und Organisationsstrukturen werden nicht beachtet!

5. Prozessverbesserung durch BPR

- BPR ist fundamentales Überdenken und radikale, dramatische Neugestaltung der Unternehmensprozesse.
- **fundamental:**
 - Warum machen wir die Dinge, die wir tun?
 - Weshalb machen wir sie auf diese Art und Weise?
- **radikal:**
 - völlige Neugestaltung,
 - Missachtung aller bestehenden Strukturen u. Verfahrensweisen,
- **dramatisch:**
 - als Resultat sind Verbesserungen um Größenordnungen angestrebt,
- **Unternehmensprozesse:**
 - Augenmerk nicht auf einzelne Aufgaben, Positionen, Menschen, Strukturen,
SONDERN auf gesamte Prozesse.

Symptome für einen Verbesserungsbedarf:

- zu viele Fälle im System,
- Durchlaufzeit im Verhältnis zur Servicezeit zu hoch (niedriger Service-Level),
- zu hohe Ressourcenauslastung,
- hohe Varianz in der Durchlaufzeit (instabiles System),
- Zahl der Organisationsbrüche,
- Zahl der Medienbrüche.

am Anfang – Zieldefinition:

- messbare Zielvorgaben, Bewertungskriterien definieren (etwa Kennziffern (z.B. Produktivität, Anz. d. Reklamationen))
- “Start from the future and work backwards!”

organisatorisches Konzept:

- Orientierung an den kritischen Geschäftsprozessen (d.h. alle Prozesse, die direkt mit der Leistungserstellung zu tun haben),
- Ausrichtung dieser Geschäftsprozesse am Kunden,
- Konzentration auf Kernkompetenzen (d.h. auf spezifische Fähigkeiten des Unternehmens, durch die es sich von den anderen Unternehmen abhebt),
- Nutzung modernster Informationstechnologie.

6. Prozessverbesserung durch CPI

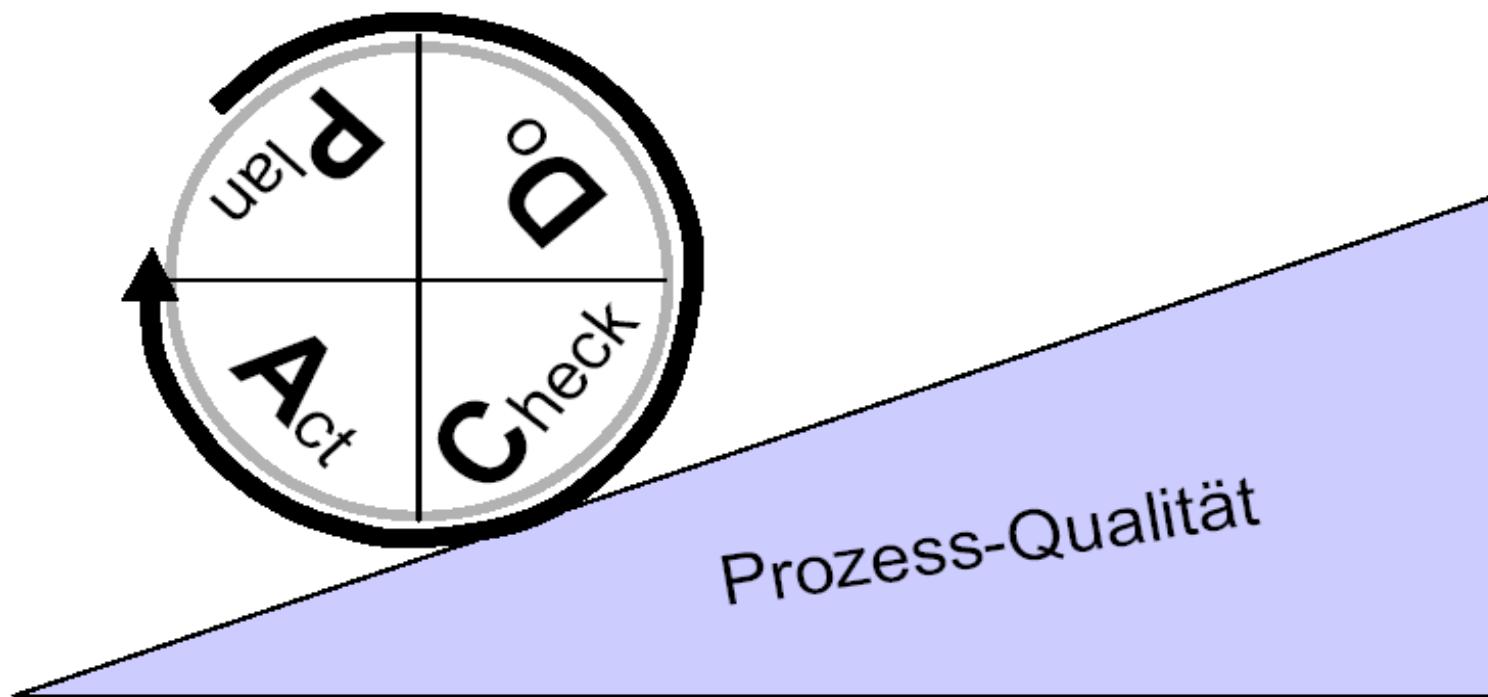
- CPI = Continuous Process Improvement,
- Anstelle eines „radikalen Durchbruchs“: Optimierung der Prozesse durch kontinuierliche, inkrementelle Verbesserung,
- CPI ist wichtiger Bestandteil des Total Quality Management - Ansatzes (TQM)
 - Kundenbezogen,
 - geht vom Qualitätsgedanken an die Problematik heran,
 - Ziel ist Erhöhung des Qualitätsbewusstseins im Unternehmen.

6. Prozessverbesserung durch CPI

- CPI als Bestandteil des TQM,
- stetige Verbesserung der Qualität des Prozesses
 - hohe Prozessqualität,
 - stetige Verbesserung der Qualität des Produktes,
 - hohe Produktqualität,
 - hohe Kundenzufriedenheit.
- jede Aktivität beinhaltet auch Messungen/Überprüfungen als Basis zur Aufdeckung von Verbesserungsmöglichkeiten,
- Verbesserungszyklen z.B. für Produktentwicklung, CRM, ...
- regelmäßige Eliminierung von Überflüssigem/Inaktuellem (z.B. unnötig gewordene Tasks,...)

6. Prozessverbesserung durch CPI

- CPI als Bestandteil des TQM,
- stetige, systematische Verbesserung der Qualität durch Verbesserungszyklen
- Plan-Do-Check-Act (PDCA) – Zyklus von Deming:



6. Prozessverbesserung durch CPI

weitere TQM-Aspekte (Einschub):

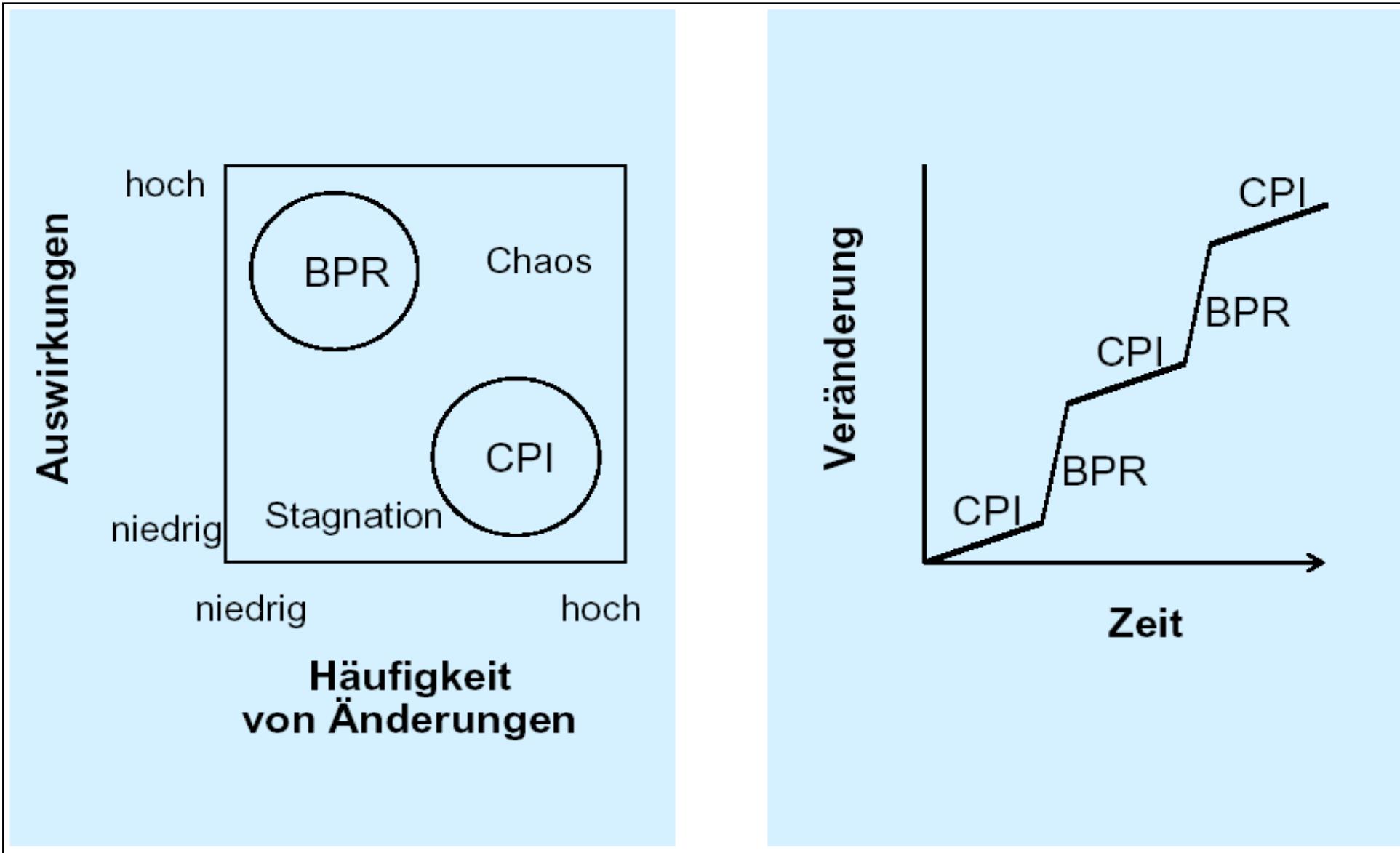
- kundenorientierte Qualität
 - Kundenzufriedenheit hat oberste Priorität im Unternehmen, nicht interne Aktivitäten,
- Schnelle Antwort auf Kundenbedürfnisse
 - kurze Produktzyklen, Einfachheit, Effizienz
- Top-Management-Einfluss
 - klare Qualitätswerte und -ziele,
 - Einführung v. Systemen / Methoden diese zu erreichen,
 - Motivation der Angestellten,
 - Leistungsindikatoren bezüglich Kundenzufriedenheit.

6. Prozessverbesserung durch CPI

weitere TQM-Aspekte (Einschub):

- Handeln nach Fakten
 - Fakten / statistische Analysen sind Basis für Planung, Überprüfung, Leistungsüberwachung und -verbesserung,
 - objektive Daten ermöglichen rationale Entscheidungen,
- Einbezug aller Angestellten
 - ständige Aus- und Weiterbildung der Angestellten,
 - Motivation zu höherer Verantwortung, Kreativität, Innovation ...
- TQM Kultur
 - offene, kooperative Unternehmenskultur,
 - Verantwortung der einzelnen Angestellten für Kundenzufriedenheit.

7. Zwei Methoden: BPR vs. CPI



7. Zwei Methoden: BPR vs. CPI

Vergleich von CPI und BPR (nach Davenport)

	Verbesserung (CPI)	Innovation (BPR)
Veränderung	schrittweise	radikal
Ausgangspunkt	existierender Prozess	völlig neuer Zustand
Änderungshäufigkeit	ständig / selten	selten / einmalig
erforderliche Zeit	kurz	lang
Vorgehensweise	bottom-up	top-down
Reichweite	begrenzt	unternehmensweit, funktionsübergreifend
Risiko	mäßig	hoch
häufiger Auslöser	statistische Kontrollen	IT
Verbesserung	meist Aufgaben	Prozess, bzgl. der gewünschten Ziele

Gliederung (vorläufig)

- Motivation
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs, BPMN, ...
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards

Gliederung (vorläufig)

- Motivation
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs, BPMN, ...
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards

Gliederung:

1. Einführung in die Modellierung,
2. Geschäftsprozess-Modellierung
3. Grundregeln der Modellierung mit Petrinetzen,
4. Petrinetze formal,
5. High-level Petrinetze,
6. Grundregeln der EPK-Modellierung,
7. Verknüpfungsoperatoren bei EPK,
8. Erweiterte EPK und ARIS,
9. EPK vs. Petrinetze,
10. BPMN.

Gliederung:

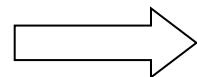
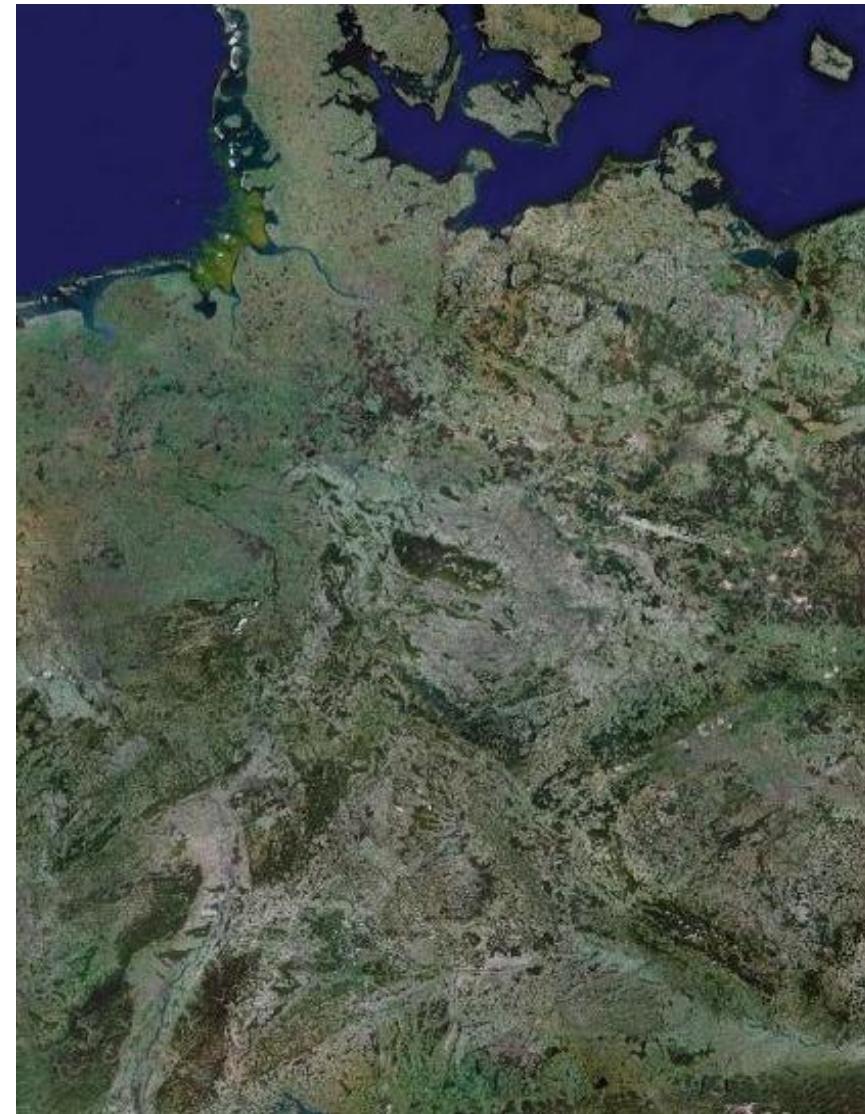
1. Einführung in die Modellierung,
2. Geschäftsprozess-Modellierung
3. Grundregeln der Modellierung mit Petrinetzen,
4. Petrinetze formal,
5. High-level Petrinetze,
6. Grundregeln der EPK-Modellierung,
7. Verknüpfungsoperatoren bei EPK,
8. Erweiterte EPK und ARIS,
9. EPK vs. Petrinetze,
10. BPMN.



Was ist ein gutes Modell?

Beispiel: Deutschland

- geringer Informationsgehalt,
- ungemessene Grafik,
- Zur Navigation nicht
verwendbar ... außer für
Astronauten ☺

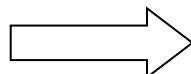


Kein gutes Modell!

1. Einführung in die Modellierung

Was ist ein gutes Modell? Beispiel: Deutschland

- klare Landesgrenze,
- Übersicht großer Städte,
- wichtigste Straßen
(Autobahnen).



gutes Modell!

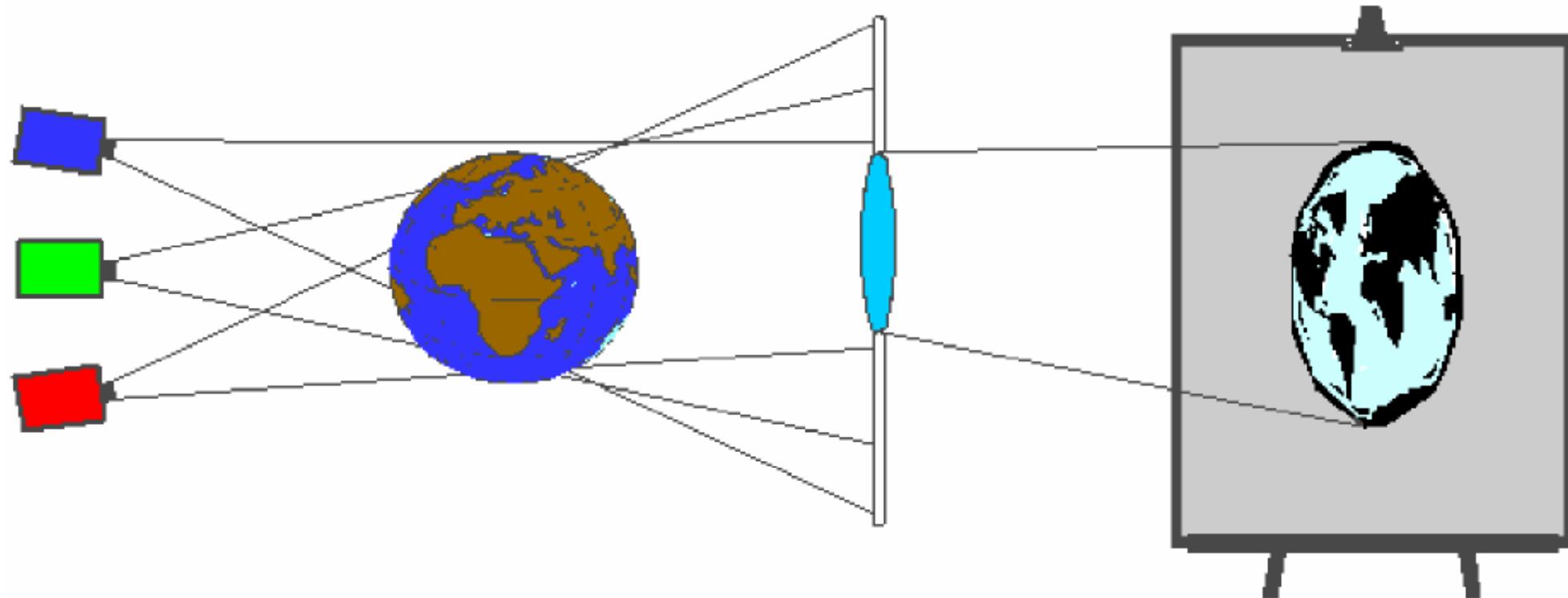


1. Einführung in die Modellierung

- **Definition:** Ein **Modell** ist ein vereinfachtes Abbild der Realität oder eines Ausschnitts der Realität. Es dient zur Beschreibung, Erklärung oder Gestaltung der Realität. Es betont einige Aspekte; ignoriert andere.
- Oft ist ein System zu komplex, um es gedanklich vollständig zu erfassen und zu untersuchen,
- Man konzentriert sich daher bei der Modellierung auf die **wesentlichen** Parameter und Wechselwirkungen des Systems,

- **Definition: Modellierung** ist die Abbildung der Realität in ein Modell auf Grundlage der Analyse und Strukturierung der Informationen über die Realität.

Prinzip der Modellierung



Methoden-
Blickwinkel

Realsystem

Methoden-
„Brille“

Modellsystem

Warum Modellierung?

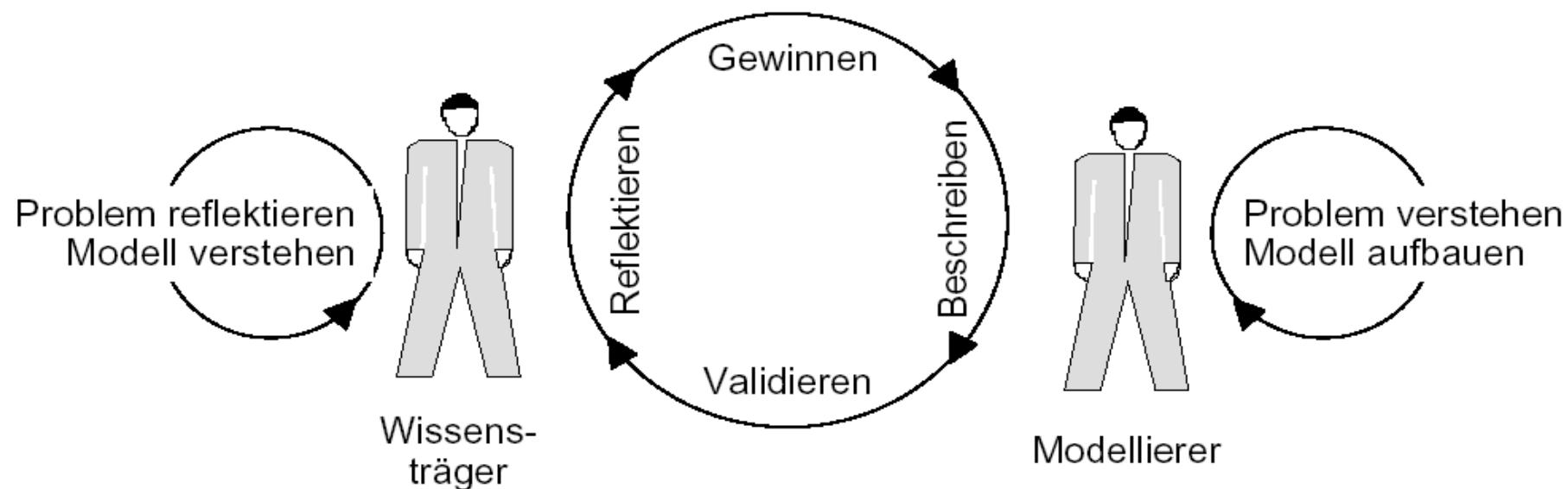
- Modelle bilden die Realität ab,
- Modelle vermitteln zwischen „Welten“,
- Modelle können ausgeführt / simuliert werden,
- Modelle können analysiert / verifiziert werden,

- Modellierung abstrahiert, strukturiert,
- Modellierung ist Kernanliegen im WfM,
- Modellierung ist ein kreativer Prozess.

1. Einführung in die Modellierung

Schema der Modellierung

- 2 Rollen, d.h. Wissensträger und Modellierer,
- Wissensträger = Person, welche das Wissen über den zu modellierenden Gegenstand oder Bereich hat,
- Modellierer = Person, die das Modell erstellt,
- In jeder Rolle kann es mehrere Personen geben, eine Person kann auch beiden Rollen gleichzeitig angehören.



Modelle, die in der Entwicklung und Nutzung von Software anzutreffen sind:

- Anforderungsmodelle,
 - Beschreibung funktionaler Anforderungen durch ein problemorientiertes Modell,
- Architekturmodelle,
 - Beschreibung einer Systemarchitektur durch ein lösungsorientiertes, konzeptionelles Modell,
- Prozessmodelle,
 - Beschreibung von Arbeitsschritten, verwendeten Ressourcen (Materialen, Personen, ...),
- Interaktionsmodelle,
 - Beschreibung der Interaktion zwischen Mensch und Rechner (sind problem- und lösungsorientiert),
- Entwurf- und Codierungsmodelle,
 - Lösungsorientierte Beschreibungen der Strukturen von Daten und Programmen,

Modelle, die in der Entwicklung und Nutzung von Software anzutreffen sind:

- Datenmodelle,
 - Beschreibung der Struktur und Zusammenhänge der Daten eines Systems,
- Funktionsmodelle,
 - Beschreibung der Funktionalität eines Systems (Funktionen und Datentransformationen),
- Verhaltensmodelle,
 - Beschreibung des dynamischen Systemverhaltens,
- Objekt- und Klassenmodelle,
 - Beschreibung der Struktur und des Verhaltens eines Systems in seinem Aufgabenumfeld,
- Qualitätsmodelle.
 - Beschreibung von Qualitätszielen und Konzepten zu Messung und Erreichung dieser Ziele.

1. Einführung in die Modellierung

Modellierung: Durch wen, für wen und wofür?

- z.B. durch:
 - Systemanalytiker,
 - Externe Berater,
 - Endanwender, Sachbearbeiter,
 - Sehr oft: Kombination obiger Personen.
- Modellierung ist stets abhängig von:
 - einer Domäne,
 - einer bestimmten Aufgabe (Herstellung vs. Verkauf).
- Daraus ergeben sich die Fragen:
 - Was ist relevant für die Modellbildung?
 - Welche Konzepte und welche Beziehungen?
 - Wie fein muss das resultierende Modell sein?

Merke: Es gibt nicht DAS richtige Modell!

Wozu Modellierung von Unternehmen und Geschäftsprozessen?

- zur Analyse und Reorganisation,
- zur Kommunikation mit Endbenutzer und Prozessverantwortlichem,
- zu Dokumentationszwecken,
- zu Entwurfs- und Wartungszwecken,
- zur Planung des Ressourcen-Einsatzes,
- als Basis für den Einsatz von Workflow-Managementsystemen bzw. von Standard-Software,
- zur Überwachung und Steuerung,
- ...

Wozu Modellierung von Unternehmen und Geschäftsprozessen?

Analyse des Prozessmodells verfolgt 3 Ziele:

Validierung

- Ist das Modell richtig bzgl. der Realität/Vorstellung?,
 - z.B. Kundenbezug, Medien- und Organisationsbrüche,

■ Verifikation

- Nachweis der Korrektheit des Geschäftsprozesses,
- Struktur (z.B. Vor- und Nachbedingungen für alle Aufgaben),
- Verhalten (z.B. Deadlocks, nie ausgeführte Aufgaben),

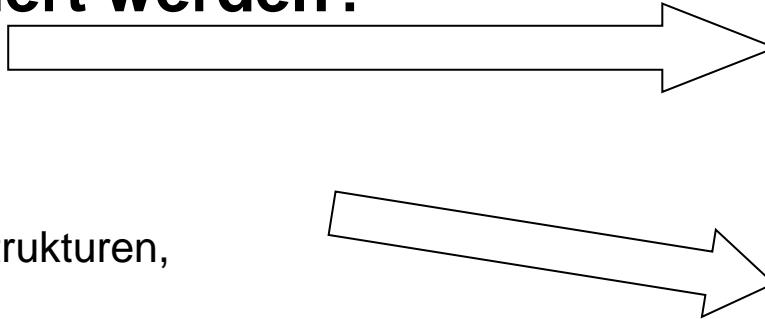
■ Leistungsbewertung

- Leistungsfähigkeit des Geschäftsprozesses,
- z.B. Durchlaufzeit, Kostenrechnung, Ressourcenauslastung.

1. Einführung in die Modellierung

Was muss modelliert werden?

- Aufgaben,
- Ablaufstrukturen,
- Ressourcen,
- Rollen und Organisationsstrukturen,
- Zeit- und Kostenaspekte,
- Datenobjekte,
- Prioritäten,
- Begriffe und Beziehungen,
- Kommunikationsstrukturen,
- Geschäftsregeln: allgemeingültige Regeln,
- Ausnahmesituationen,
- Qualitätsanforderungen an zu erzeugende Produkte,
- Sicherheitsanforderungen.



- Sequenz,
- Alternative,
- Wiederholung,
- Parallelität,
- Unabhängigkeit.

- Zuständigkeiten,
- Verantwortlichkeiten,
- Kompetenzen.

Anforderungen an Modellierungssprachen

- Ausdrucksmächtigkeit,
 - alle relevanten Aspekte müssen modellierbar sein,
 - Adäquatheit/Angemessenheit der Modellierungskonstrukte.
- Erweiterbarkeit,
 - später benötigte Konstrukte müssen hinzuzufügen sein,
- dynamische Anpassbarkeit,
 - zur Reaktion auf veränderte Marktbedingungen,
- Wiederverwendbarkeit,
 - zur Vermeidung aufwändiger Neuentwicklungen,
- Offenheit,
 - zur Integration von existierenden und neuen Anwendungssystemen,
- Einfachheit, Verständlichkeit,
 - Leicht zu lernen, Leicht zu benutzen,
- Formalisierungs- bzw. Präzisierungsgrad
 - flexible Anpassbarkeit an das Ziel der Modellierung, die Zielgruppe des Modells.

Anforderungen an Modellierungssprachen

- Visualisierungsmöglichkeiten,
 - graphische Darstellung (leichte Handhabbarkeit, Lesbarkeit, Abstraktion),
 - unterschiedliche Sichten, Modularisierbarkeit, Detaillierungsgrad.
- Entwicklungsunterstützung,
 - methodische Unterstützung für die Modellierung,
 - Werkzeugunterstützung.
- Analysierbarkeit, Ausführbarkeit/Simulierbarkeit,
 - Validierung, Verifikation, Leistungsbewertung,
 - formale Repräsentation,
 - Prüfung syntaktischer Eigenschaften (isolierte Elemente, Zyklen, ...),
 - Konsistenz des Modells,
 - Analyse anwendungsbezogener Aspekte (Durchlaufzeiten, Reaktionszeit, ...),
 - inhaltliche Richtigkeit (entspricht Modell der Realität).
- Unabhängigkeit von Herstellern.

Adäquatheit der Modellierungskonstrukte

- abhängig zum einen von der Sichtweise:
 - Kunde oder Anwender,
 - Analytiker,
 - Designer,
 - Programmierer.
- zum anderen vom Zweck:
 - zur Anforderungsanalyse,
 - zu Entwurfszwecken,
 - zur Codierung,
 - zur Dokumentation.
- verschiedene Abstraktionsgrade der Modellierung, z.B. bei Datenmodellen:
 - konzeptuelles Modell,
 - logisches Modell.

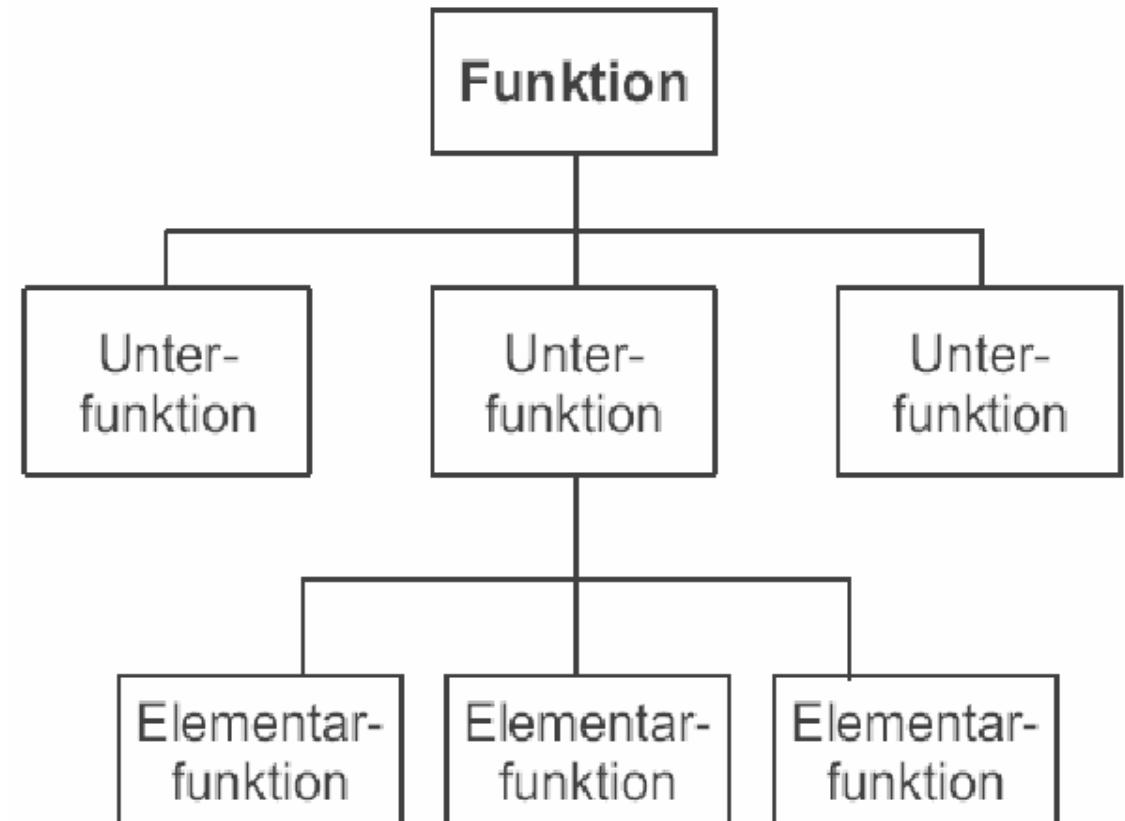
Unterschiedliche Ansätze zur Modellierung

- verschiedene Modellierungsansätze
 - funktionale,
 - objektorientierte,
 - agentenorientierte,
 - prozessorientierte.



Funktionale Modellierung

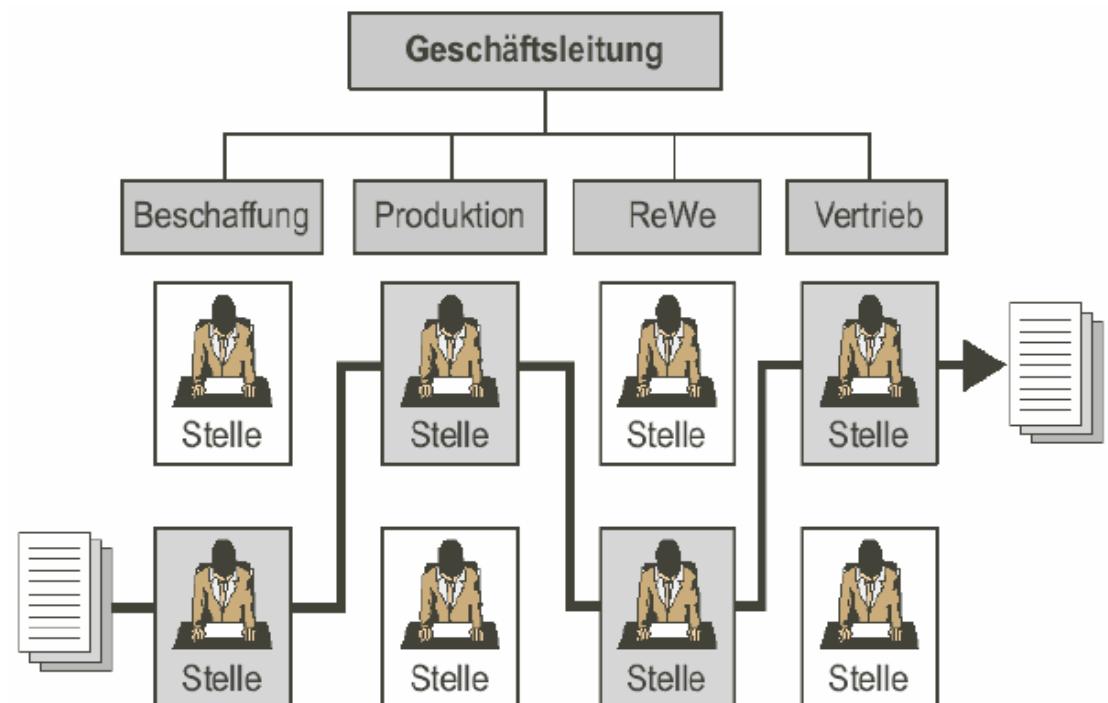
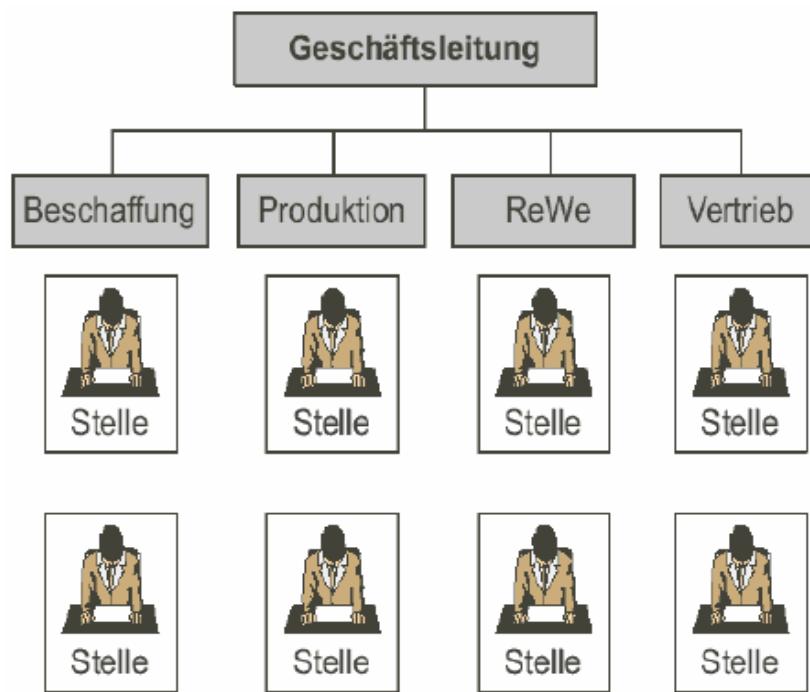
- Beschreibung der Welt durch funktionale Blöcke,
- Hierarchische Verfeinerung dieser Blöcke,
- Zuordnung von Daten und Ressourcen zu Blöcken,
- Verknüpfung der Blöcke durch Funktionsaufrufe.



1. Einführung in die Modellierung

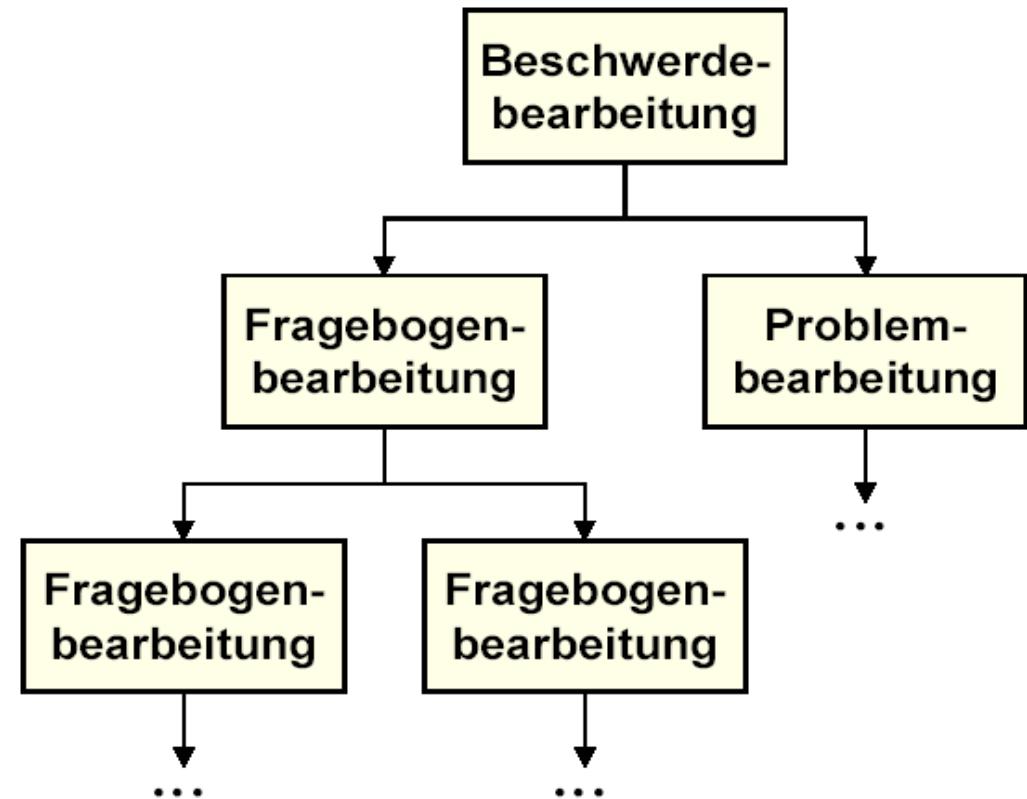
Funktionale Modellierung

- Traditionell statische Aufbau-Organisation im Unternehmen.
- Funktionen liegen quer zur Aufbau-Organisation.



Funktionale Modellierung (Eigenschaften)

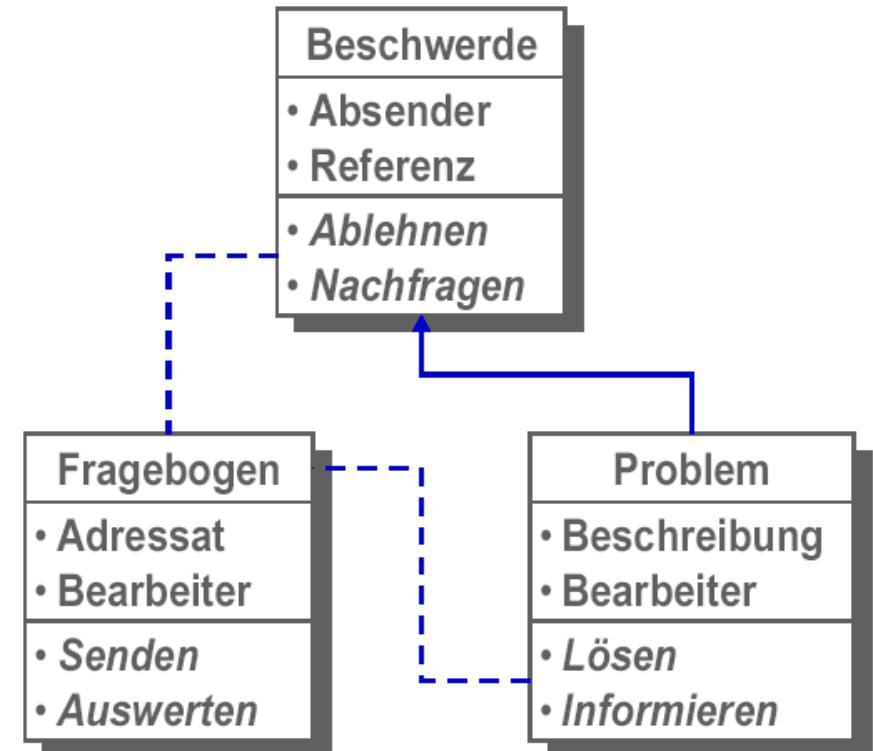
- hohe Arbeitsteilung,
- viele Schnittstellen in der Bearbeitungsfolge,
- lange Bearbeitungszeiten,
- hoher Koordinationsbedarf,
- starre Hierarchiegrenzen und Ablaufgrenzen





Objektorientierte Modellierung

- Beschreibung der Welt durch Objekte mit
 - Eigenschaften,
 - Fähigkeiten,
- Konstruktion komplexer Objekte aus einfachen,
- Spezialisierung / Generalisierung von Objekten,
- Bereitstellung von Schnittstellen,
- Kapselung der Interna.

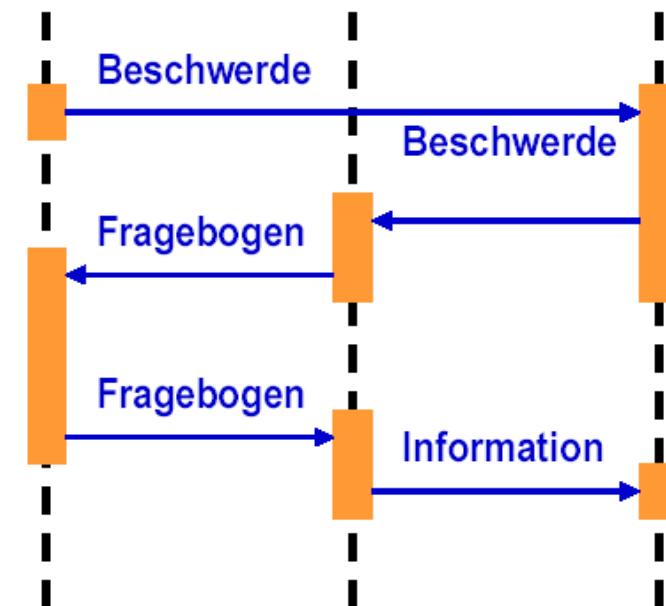




Agentenorientierte Modellierung

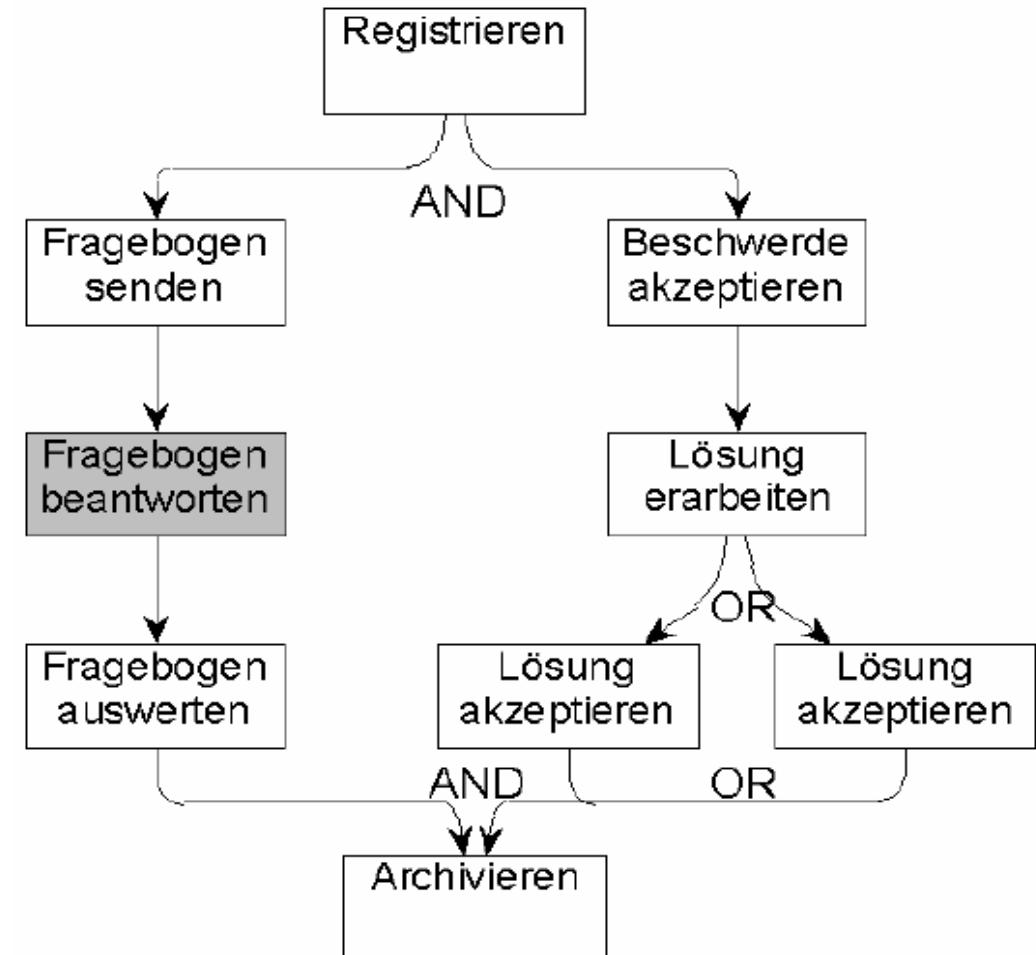
- Beschreibung der Welt durch Agenten mit:
 - Fähigkeiten,
 - Wissen,
 - Zielen.
- Dezentrale Funktionalität und Kontrolle,
- Strukturierung durch Sub-Agenten,
- Interaktion durch Kommunikation.

Kunde Mitarb. Zentrale



Prozessorientierte Modellierung

- Beschreibung der Welt durch Aktivitäten und deren Ordnung
- Hierarchische Verfeinerung der Aktivitäten,
- Modellierung von Daten und Ressourcen als Bedingungen,
- Einbindung der Umgebung mit externen Aktivitäten.



Prozessorientierte Modellierung (Vorteile)

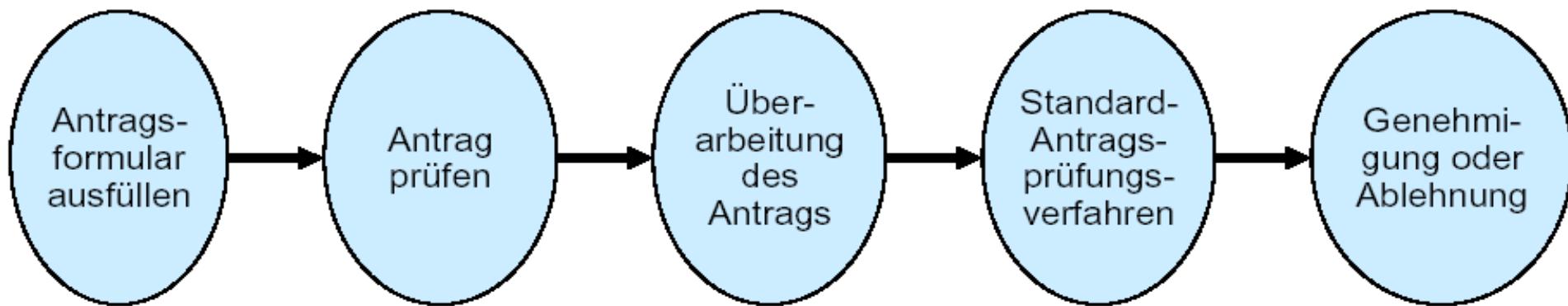
- Ganzheitliche Betrachtung der Prozesse,
- Trennung von Prozesslogik und Applikationen,
- Fokussierung auf dynamische Aspekte,
- Simulierbarkeit operationaler Modelle,
- Integration von Informationen und Ressourcen,
- Quasi-Standard in der WFMC.

Überblick

- Flussdiagramme (Flowcharts)
- Activity-Diagramme,
- Datenflussdiagramme (DFD),
- Use-Case-Diagramme,
- Transitionssysteme, Zustandsdiagramme (State Chart),
- Warteschlangen-Modelle, Markov-Ketten,
- Prozess-Algebren,
- Interaktionsdiagramme,
- Ereignisgesteuerte Prozessketten (EPK),
- Petrinetze.

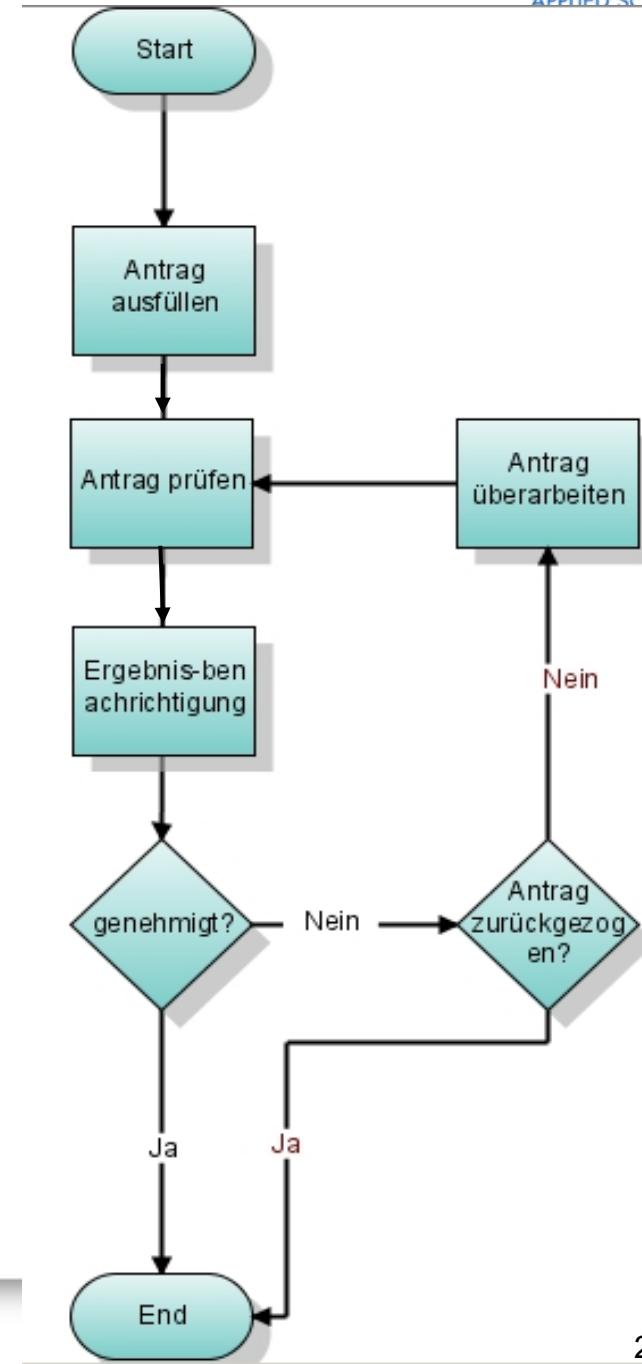
2. Geschäftsprozess-Modellierung

Beispiel: Urlaubsbeantragung



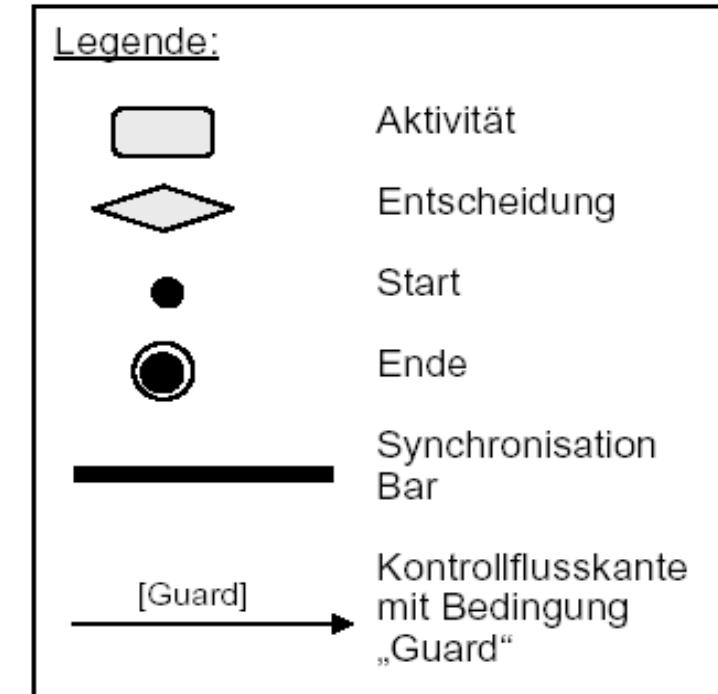
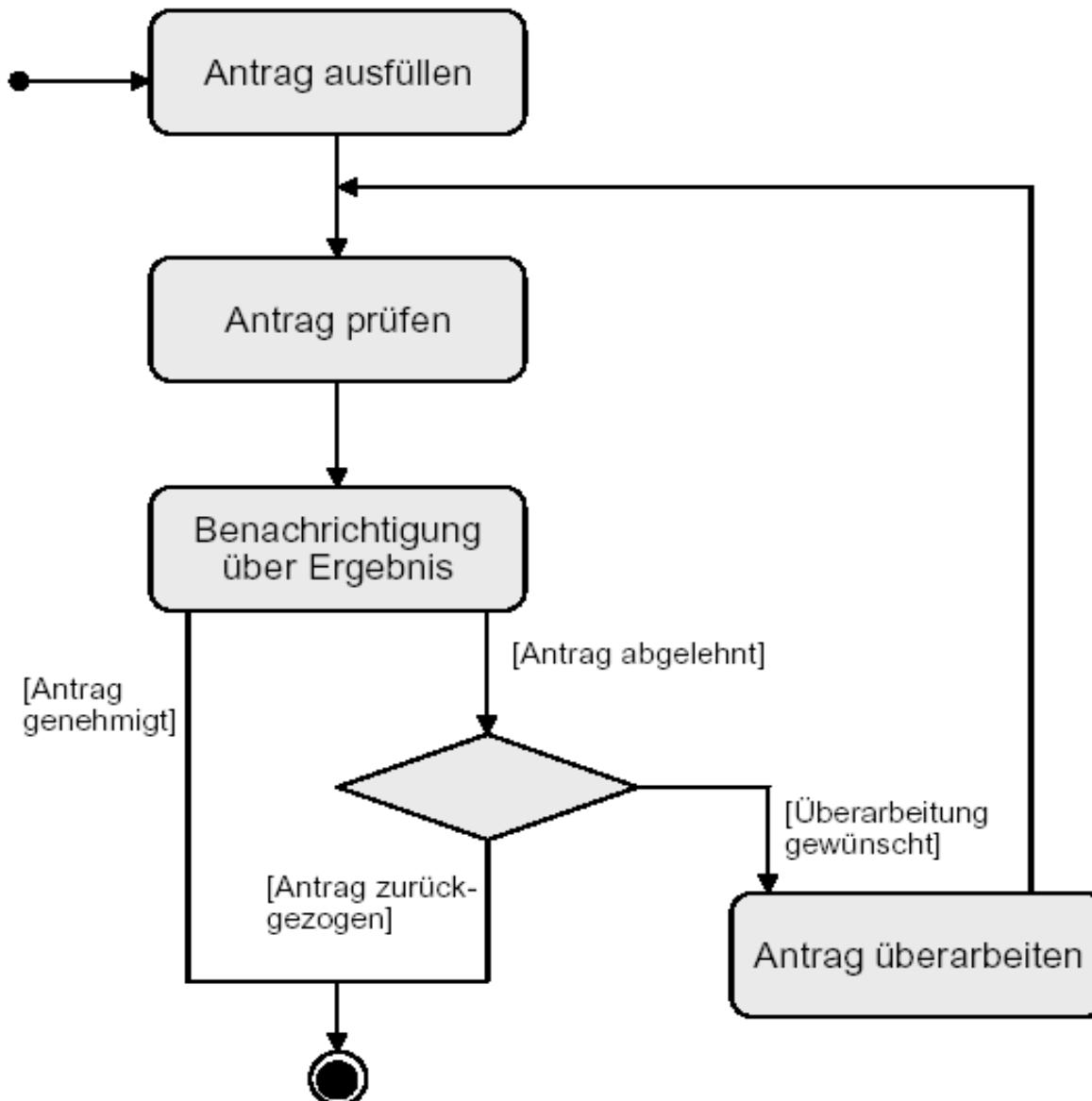
2. Geschäftsprozess-Modellierung

Beispiel: Urlaubsantrag als *Flowchart*
(DIN 66001)



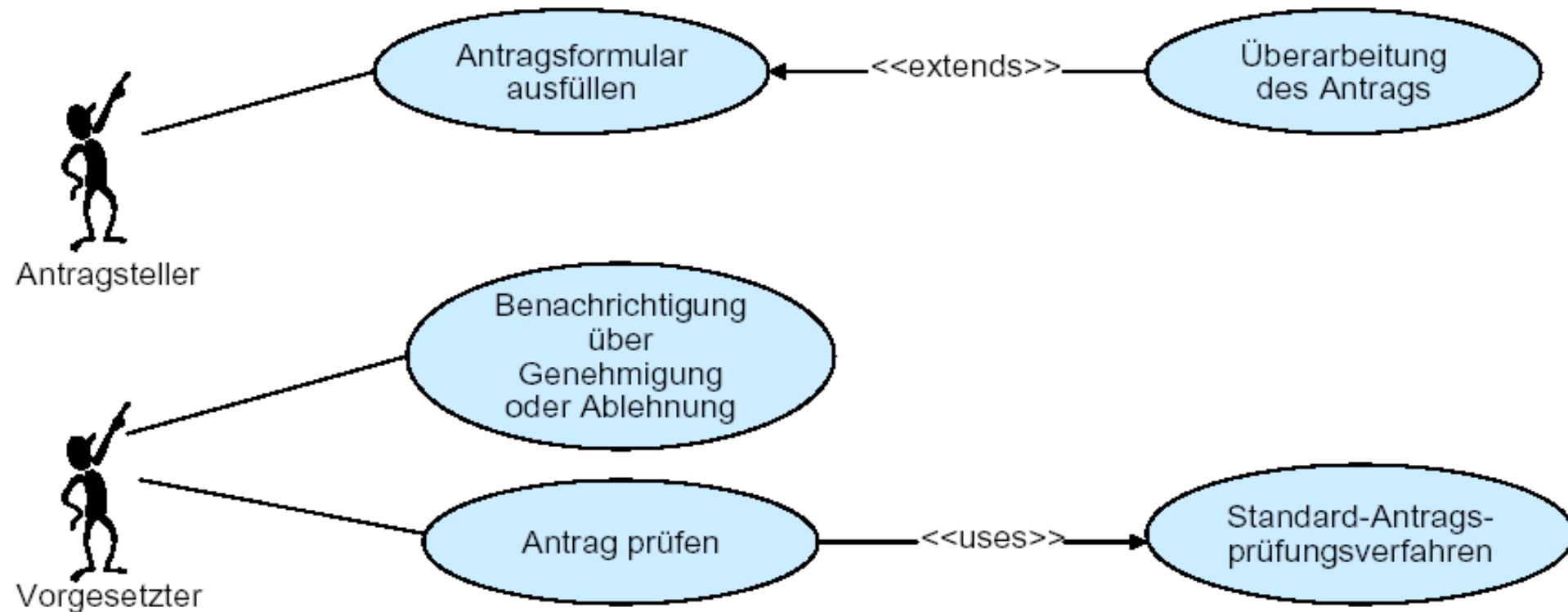
2. Geschäftsprozess-Modellierung

Beispiel: Alternativrealisierung durch ein *Activity-Diagramm*



2. Geschäftsprozess-Modellierung

Beispiel: Urlaubsbeantragung als *Use Case Diagramm*



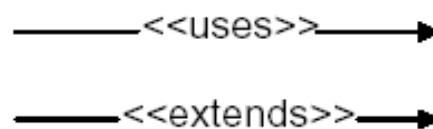
Legende:



Actor
(und/oder Rollen)



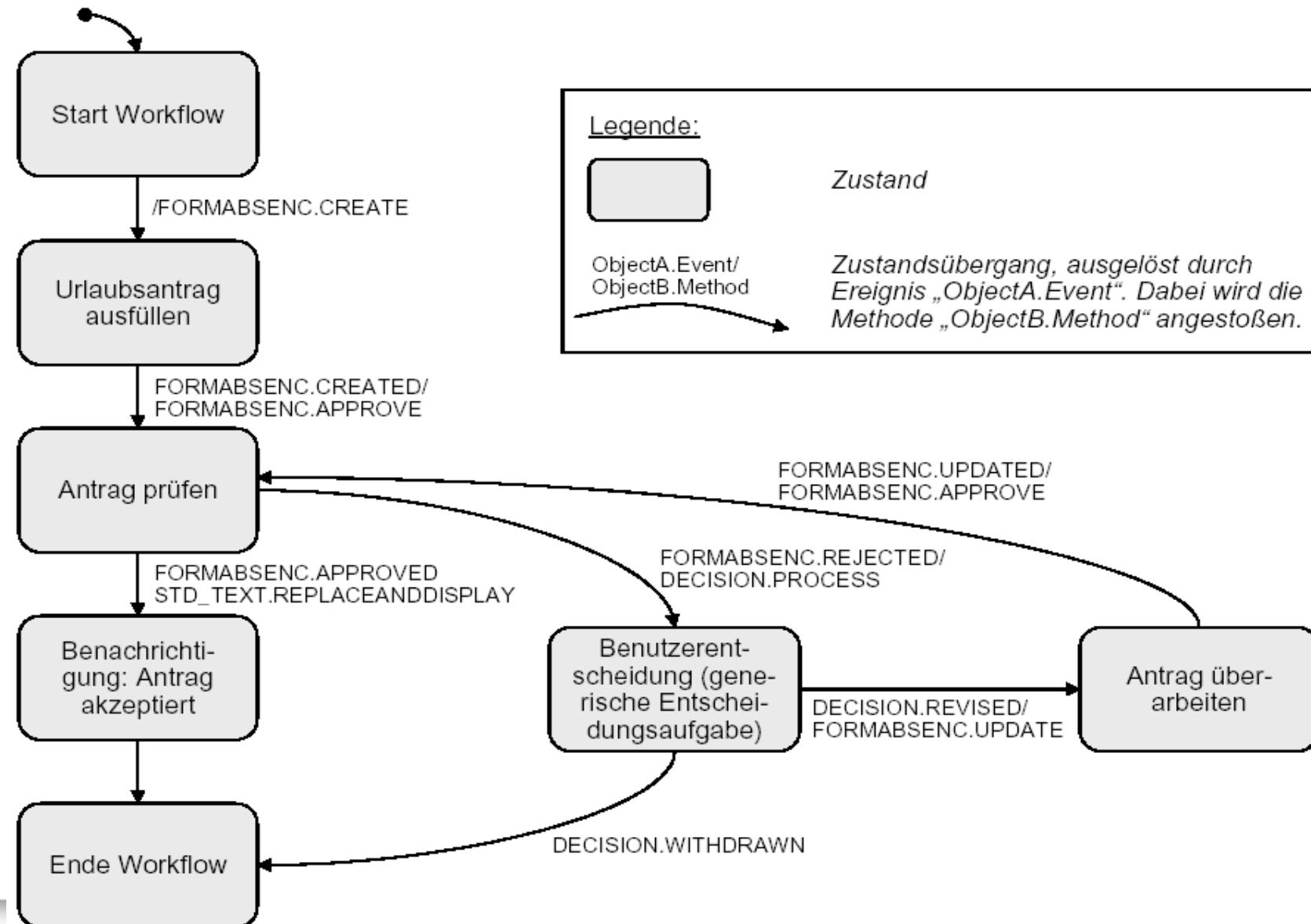
Use-Case



Spezielle Use Case
Relationen

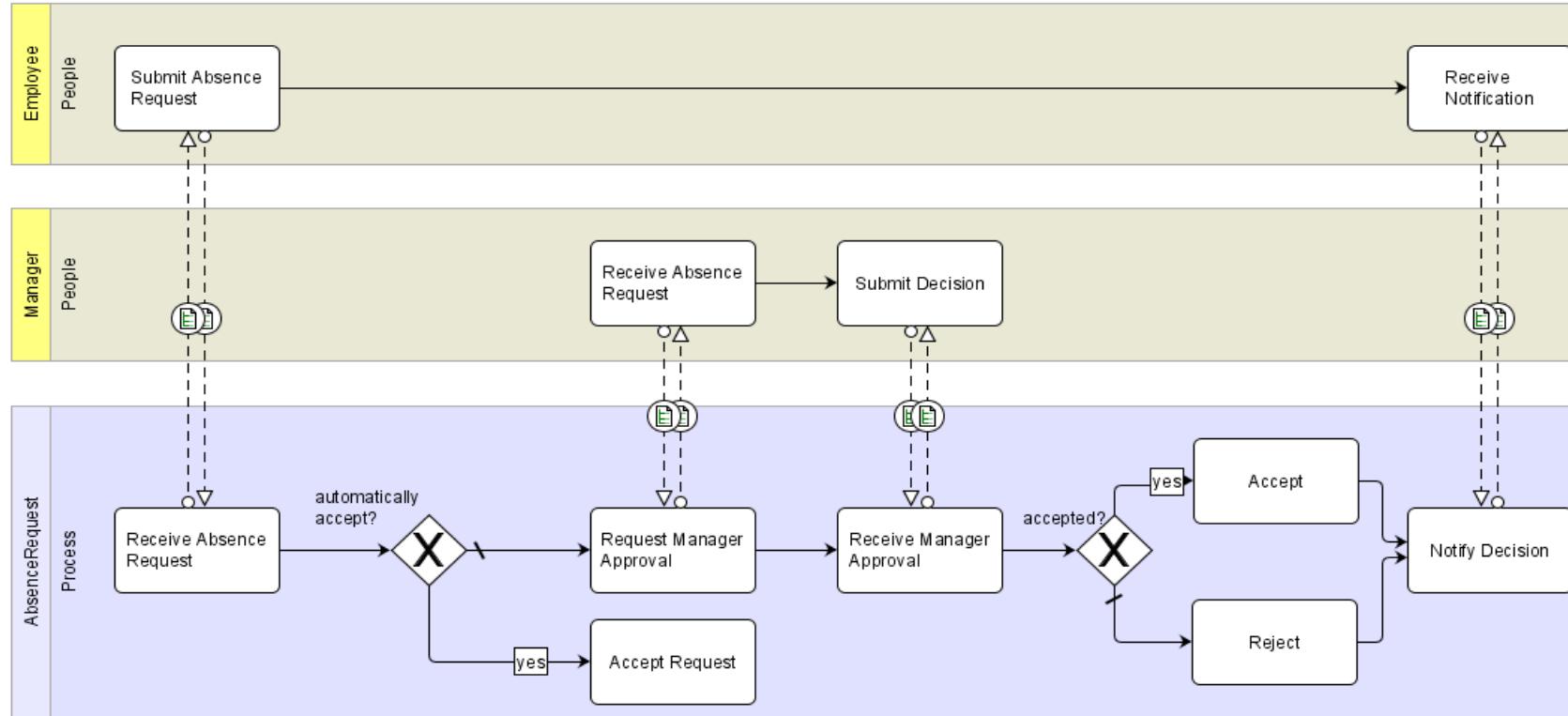
2. Geschäftsprozess-Modellierung

Beispiel: Urlaubsbeantragung als *State Chart*



2. Geschäftsprozess-Modellierung

Beispiel: Urlaubsantrag, modelliert in *BPMN* (mit intalio BPMS)



- **Flow Objects** - Knoten in den Geschäftsprozessdiagrammen,
- **Connecting Objects** - verbindende Kanten in den Geschäftsprozessdiagrammen,
- **Swimlanes** - die Bereiche, mit denen Akteure und Systeme dargestellt werden,
- **Artifacts** - weitere Elemente wie Data Objects, Groups und Annotations.

Geschäftsprozess-Management

Prof. Dr.-Ing. Andreas Ittner

Email: ittner@hs-mittweida.de

WWW: www.hs-mittweida.de/~ittner

Tel.: +49(0)3727-58-1288

Mob.: +49(0)177-5555-347

Gliederung (vorläufig)

- Motivation
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs, BPMN, ...
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards

Gliederung:

1. Einführung in die Modellierung,
2. Geschäftsprozess-Modellierung
3. Grundregeln der Modellierung mit Petrinetzen,
4. Petrinetze formal,
5. High-level Petrinetze,
6. Grundregeln der EPK-Modellierung,
7. Verknüpfungsoperatoren bei EPK,
8. Erweiterte EPK und ARIS,
9. EPK vs. Petrinetze,
10. BPMN.

3. Grundregeln der Modellierung mit Petrinetzen

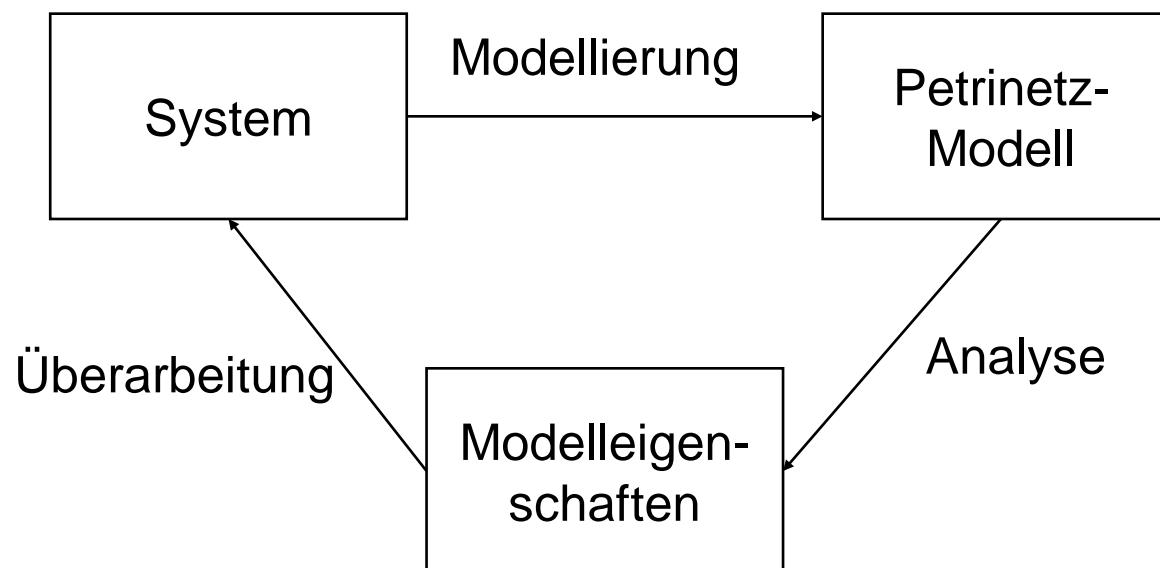


Was sind Petrinetze?

- Ursprung: Dissertationsschrift "Kommunikation mit Automaten" von Carl Adam Petri (1962),
- Seither: mehr als 10.000 Arbeiten auf dem Gebiet,
- bis 1985: hauptsächlich von Theoretikern benutzt,
- seit Mitte der 80er Jahre: vermehrter Einsatz in praktischen Anwendungen,
- Gründe:
 - Einführung der High-Level-Netze,
 - Entwurf von Werkzeugen.
- Petri Nets World: <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>
- Petrinetz-Applet:
http://wwwis.win.tue.nl/~wvdaalst/workflowcourse/pn_applet/pn_applet.htm

Warum Petrinetze?

- dienen zur Modellierung, Analyse, Simulation von dynamischen Systemen mit nebenläufigen und nichtdeterministischen Systemen,
- erlauben die Beschreibung von Kontroll- und Datenfluss.

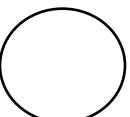
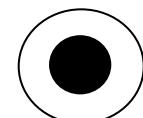


Gründe

- graphischer Formalismus,
- formale Syntax und Semantik,
- explizite Darstellung von Zuständen,
- herstellerunabhängig,
- viele Analyseverfahren und –werkzeuge.

3. Grundregeln der Modellierung mit Petrinetzen

Ein Petrinetz (gerichteter, bipartiter Graph) besteht aus

- Stellen (passive Komponenten), 
- Transitionen (aktive Komponenten), oder 
- Verbindungen/Kanten zwischen Stellen und Transitionen, 
- Marken in Stellen (Ablaufmechanismus) 

3. Grundregeln der Modellierung mit Petrinetzen

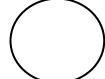
- Stellen (haben eine Kapazität*)
 - Bedingungen, Medien,
 - Materialbehälter, Datenspeicher,
 - Puffer, Nachrichtenkanäle, ...
- Transitionen
 - Ereignisse, Aktionen, Handlungen,
 - Transporte, Transformationen,
 - Anweisungen, Programme, ...
- Verbindungen/Kanten
 - Vor- und Nachbedingungen von Aktivitäten,
 - Start und Ziel von Transporten,
 - Eingabe und Ausgabe von Programmen, ...

* haben Stellen die Kapazität 1, sprechen wir von einem Bedingungs-/Ereignis-Netz

3. Grundregeln der Modellierung mit Petrinetzen

- Marken
 - Zustände einer Bedingung, Gültigkeit von Bedingungen,
 - Füllungsgrad von Speichern,
 - Daten auf Datenträgern,
 - Nachrichten in Puffern, ...
- Markierungen
 - lokale Zustände,
 - Gesamtzustände.

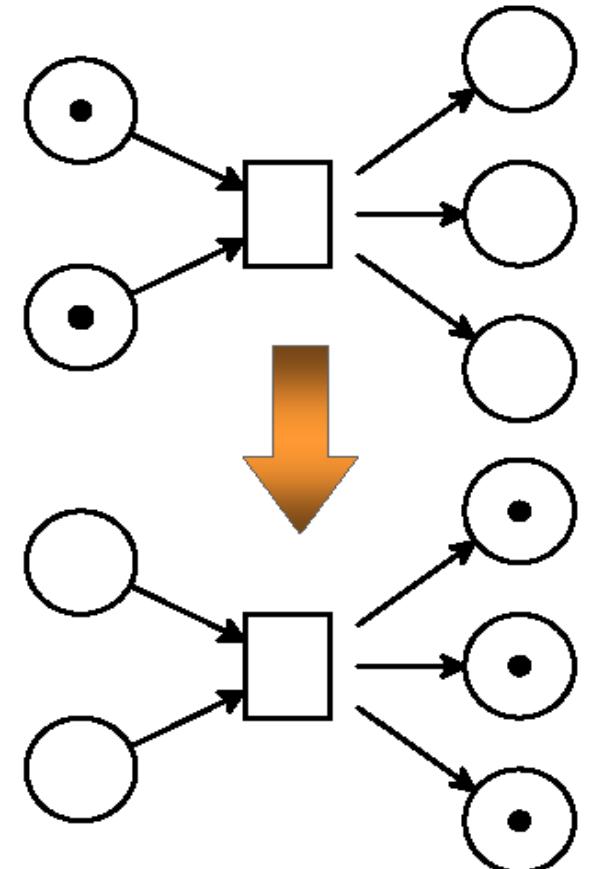
3. Grundregeln der Modellierung mit Petrinetzen

- Eine Bedingung ist entweder
 - erfüllt  oder
 - nicht erfüllt 
- Ein Ereignis hat
 - Vorbedingungen $(\text{O} \rightarrow \square)$ und
 - Nachbedingungen $(\square \rightarrow \text{O})$

3. Grundregeln der Modellierung mit Petrinetzen

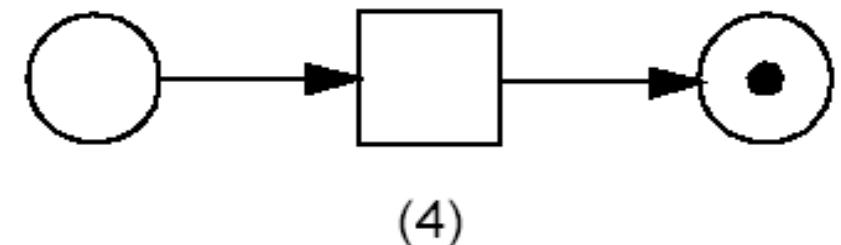
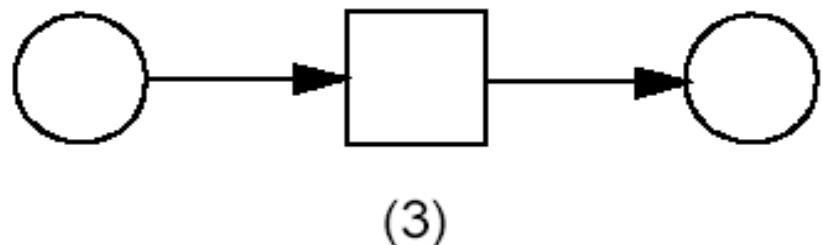
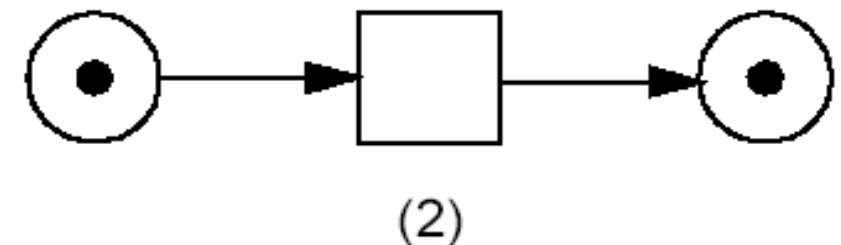
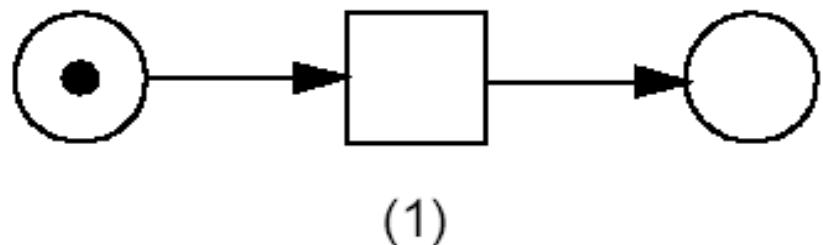
Das Verhalten

- Ein Ereignis ist aktiviert, wenn
 - alle Vorbedingungen erfüllt und
 - alle Nachbedingungen nicht erfüllt sind.
- Ein Ereignis tritt ein („schaltet“)*, d.h.
 - alle Vorbedingungen werden auf nicht erfüllt und
 - alle Nachbedingungen auf erfüllt gesetzt.



* Oft spricht man auch davon, dass eine Transition „feuert“.

Schaltbare Ereignisse in einem Bedingungs-/Ereignis-Netz?



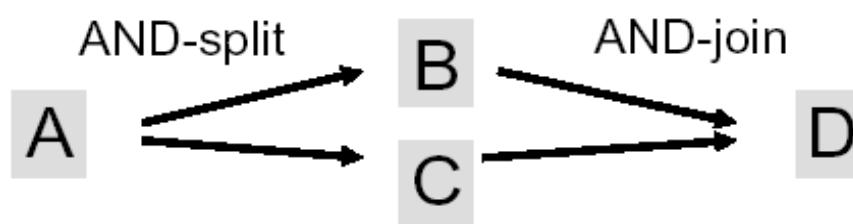
Kurze Erinnerung: Routing von Fällen:

- A, B, C, D - Aufgaben

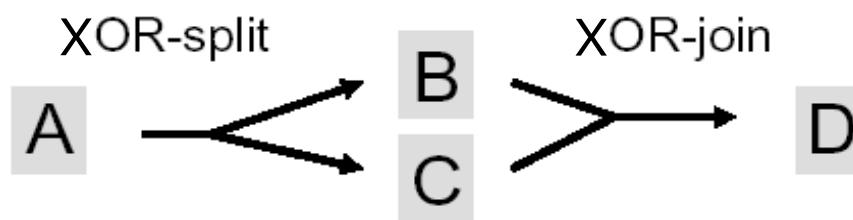
- sequentiell



- parallel



- alternativ



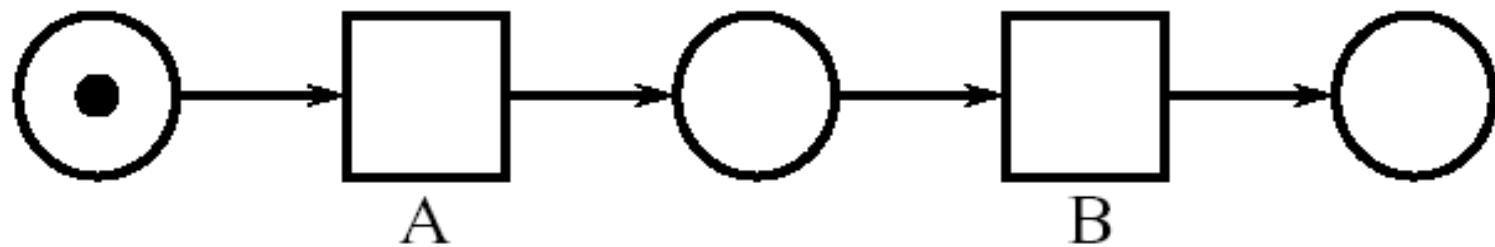
- iterativ



3. Grundregeln der Modellierung mit Petrinetzen

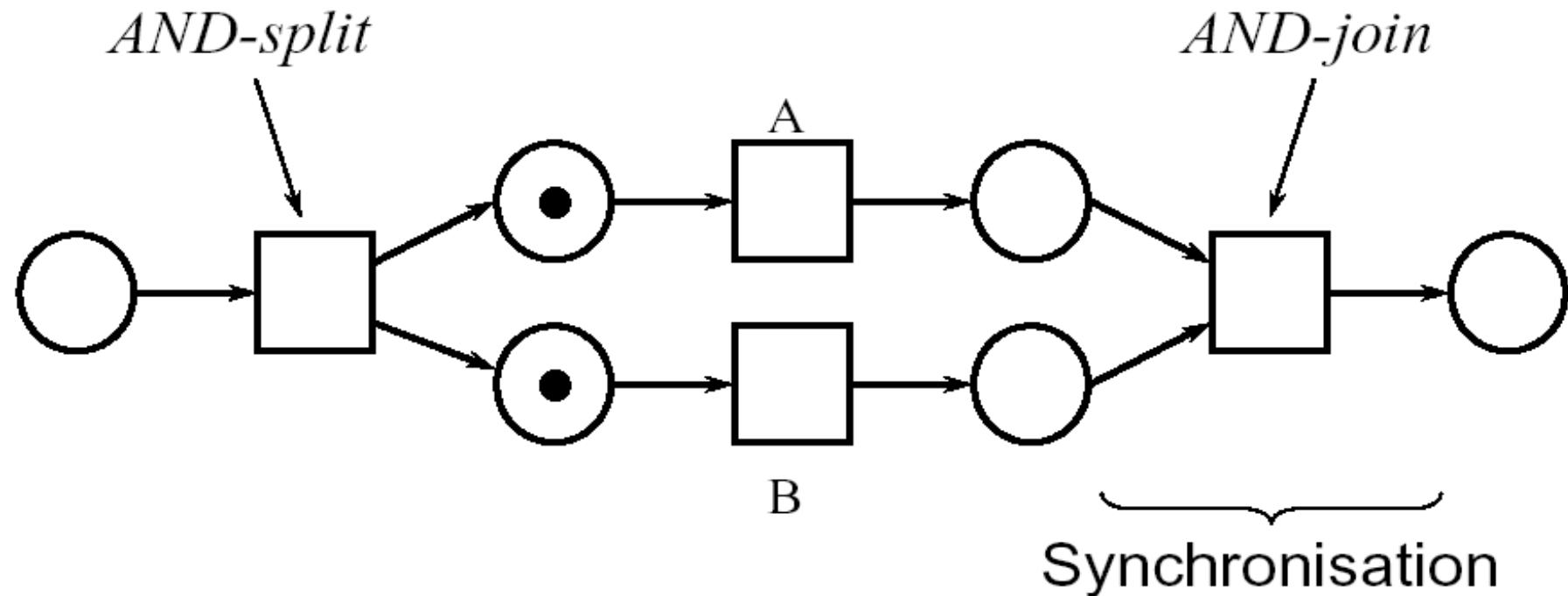
Sequentielles Routing

„Erst A, dann B“



3. Grundregeln der Modellierung mit Petrinetzen

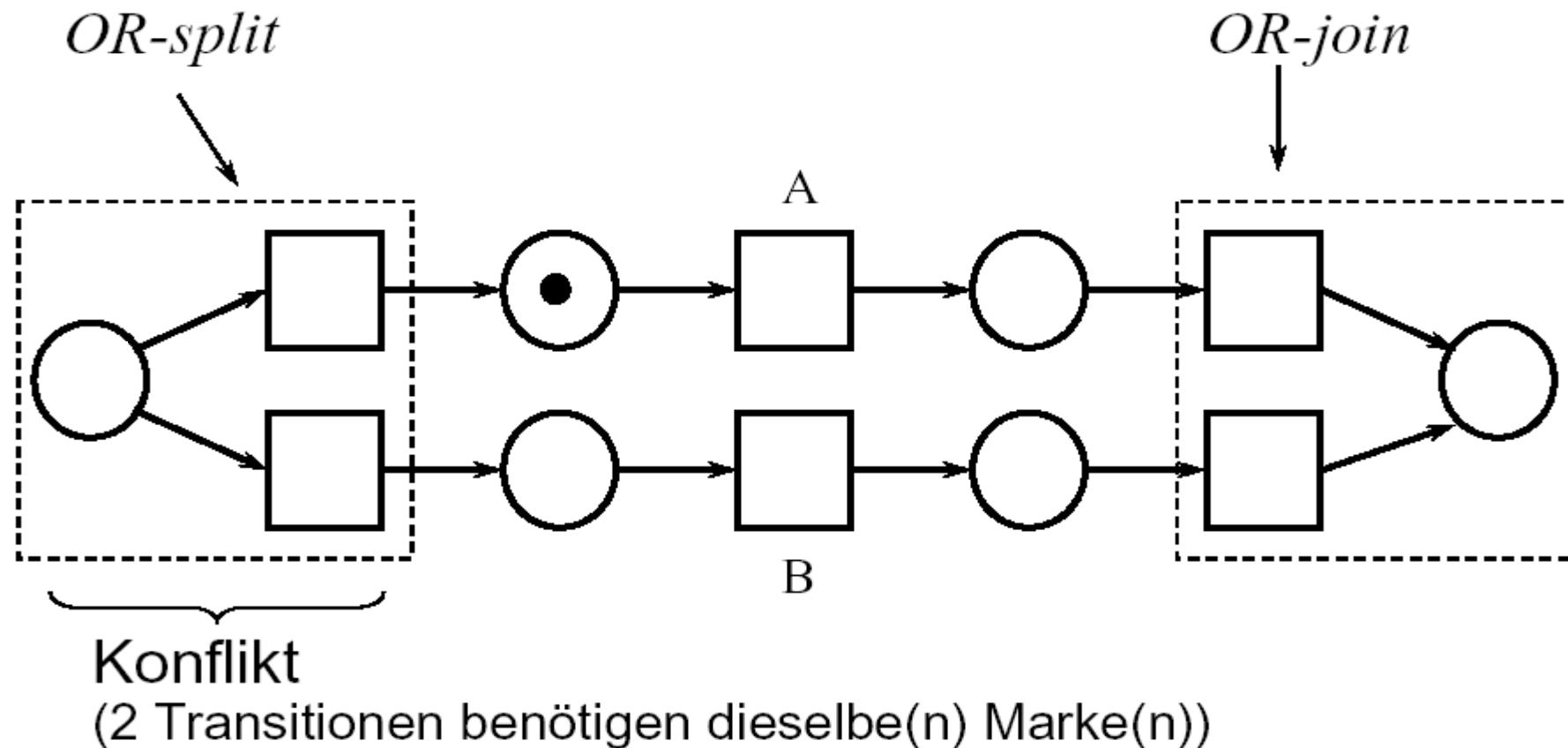
Paralleles Routing „A und B nebenläufig“



3. Grundregeln der Modellierung mit Petrinetzen

Auswahl (I) „A oder B“

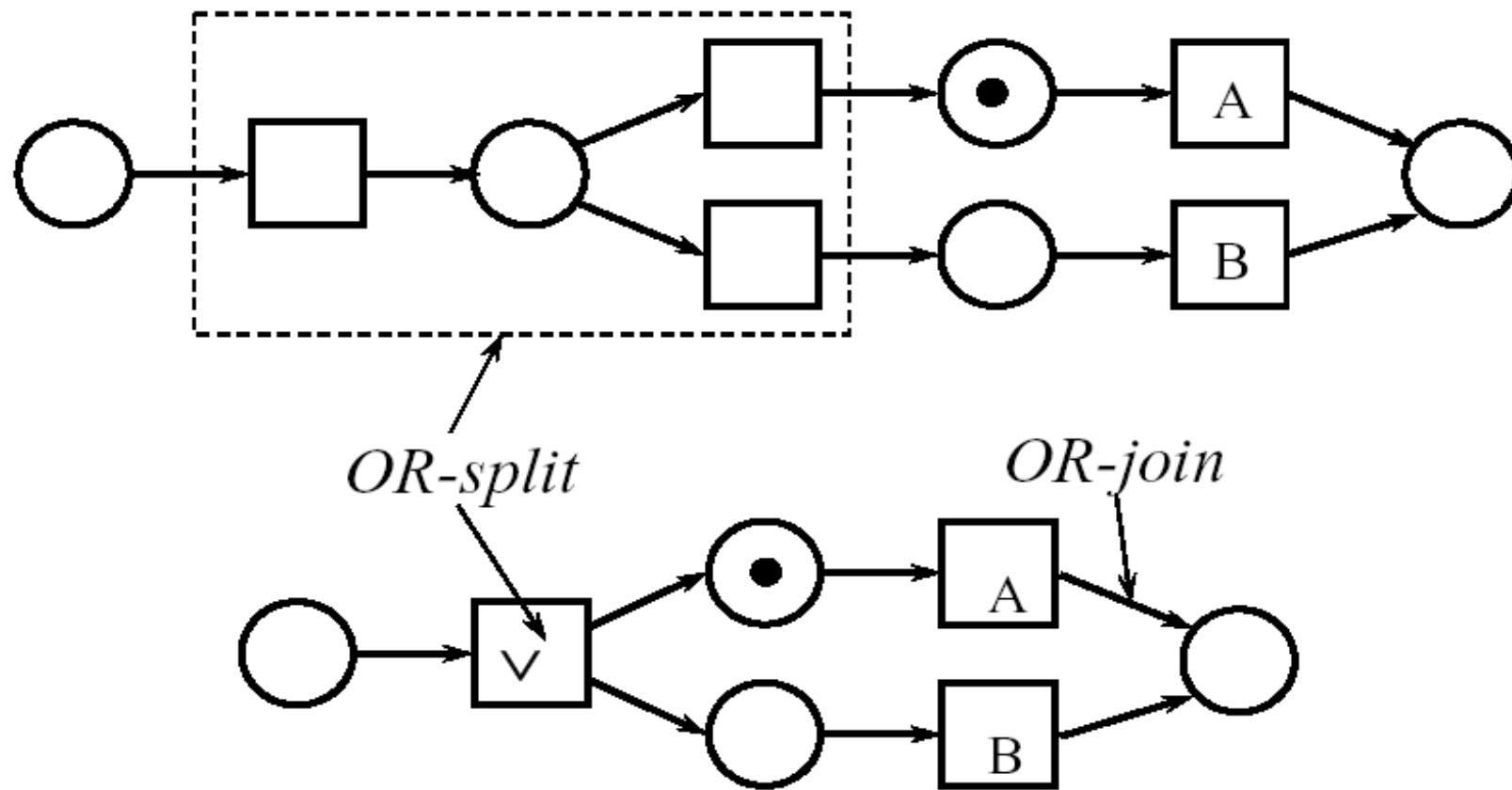
Implizite Auswahl: hängt von A und B ab (d.h. die genaue Verzweigung steckt implizit in den Transitionen)



3. Grundregeln der Modellierung mit Petrinetzen

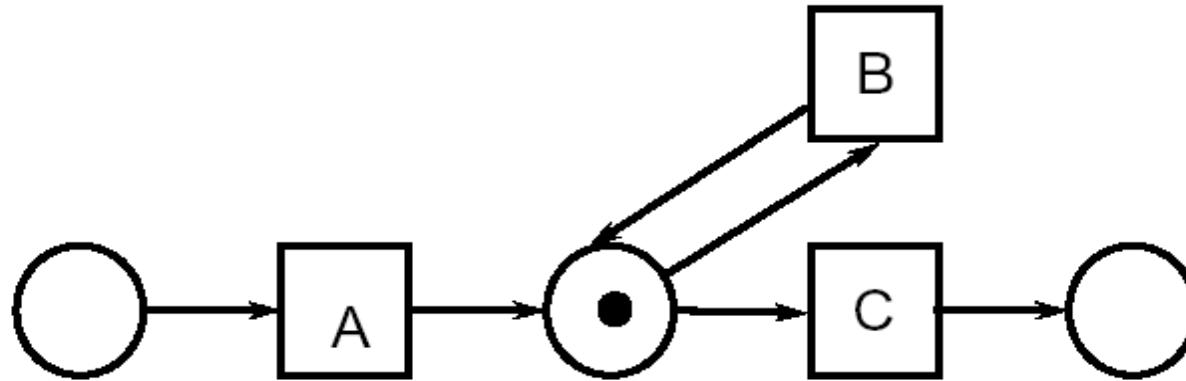
Auswahl (II)

Modellierung expliziter Auswahl:
Explizite Auswahl hängt nicht von A und B ab.

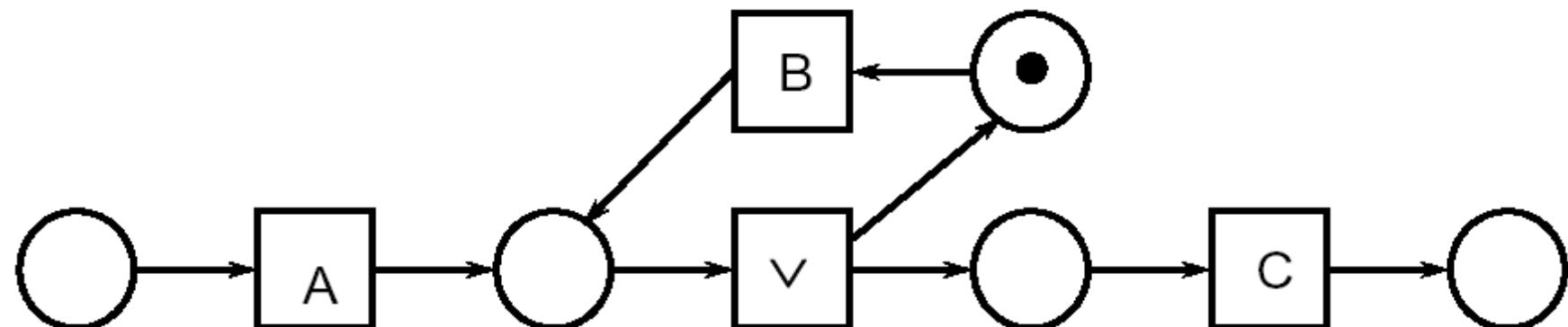


3. Grundregeln der Modellierung mit Petrinetzen

Iteration

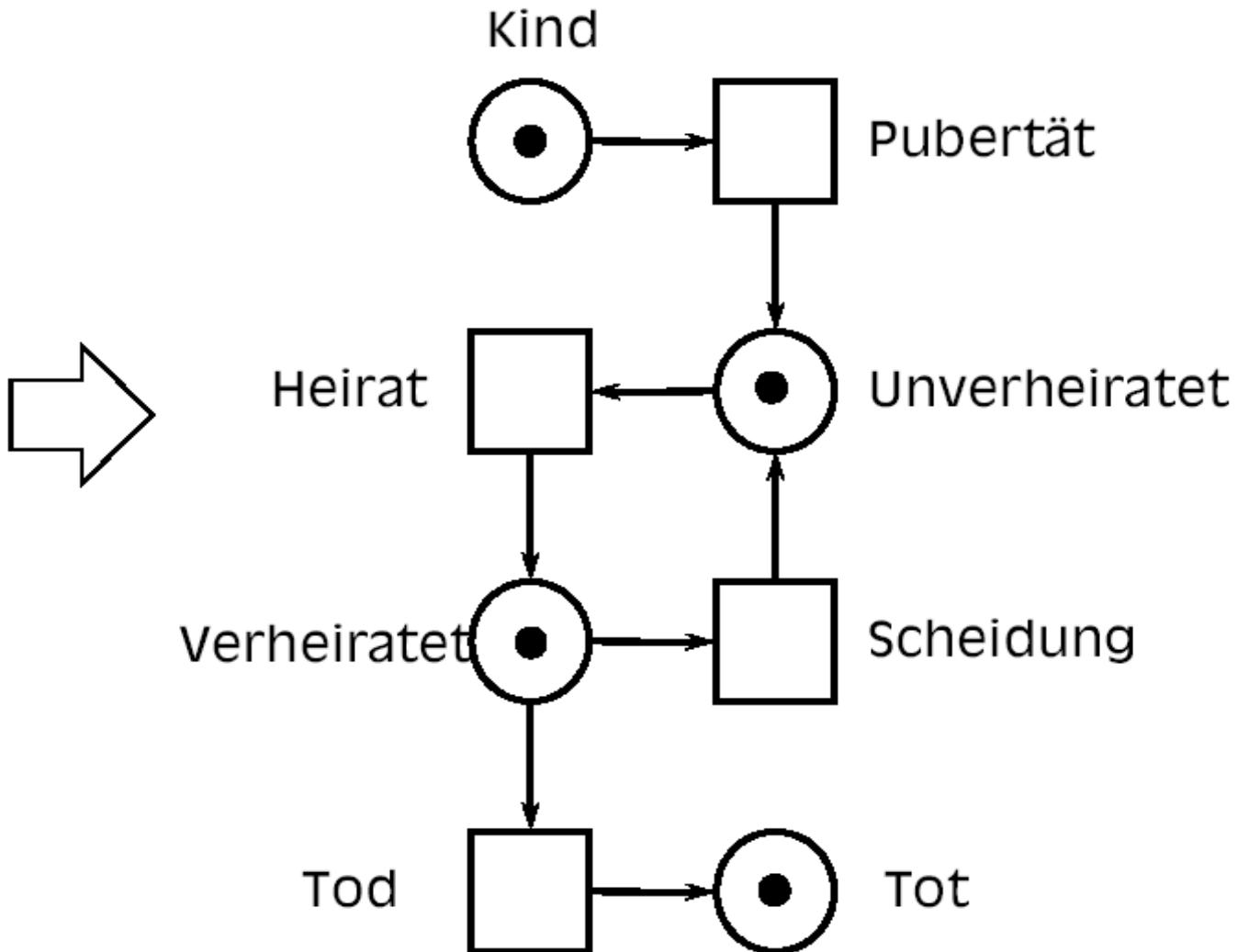


oder



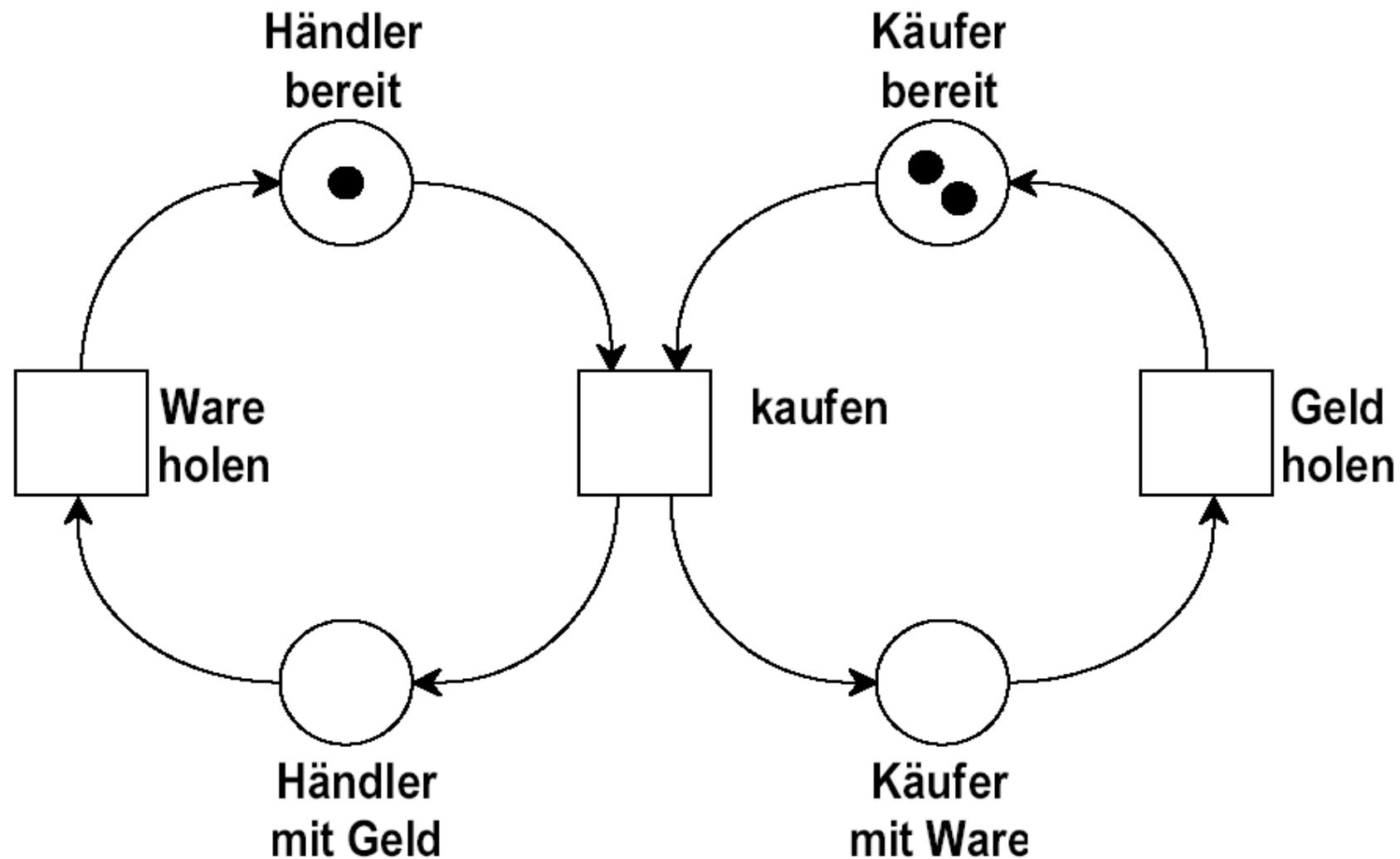
3. Grundregeln der Modellierung mit Petrinetzen

Beispiel: Lebenszyklus



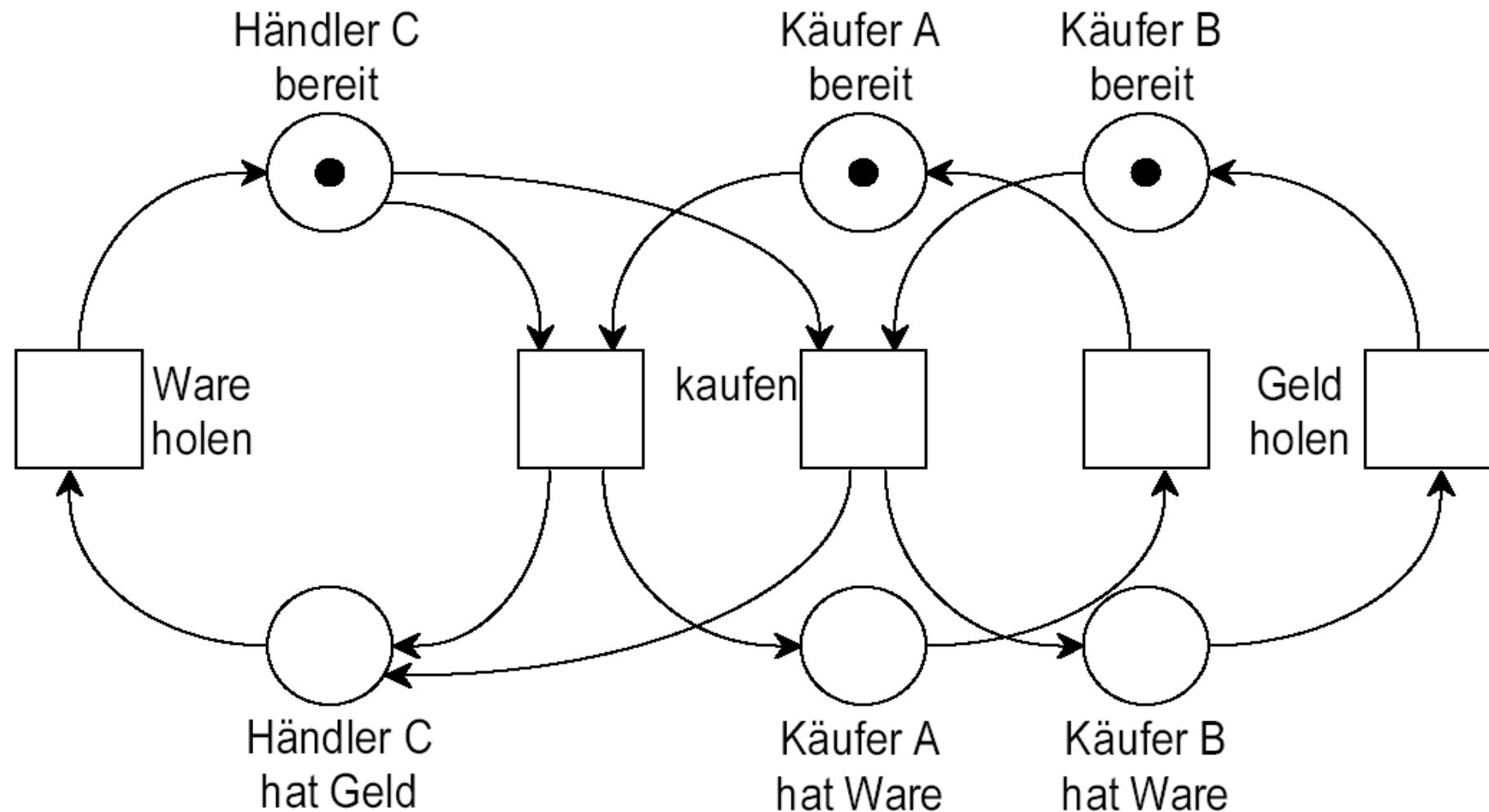
3. Grundregeln der Modellierung mit Petrinetzen

Beispiel: Ein Händler, zwei ununterscheidbare Käufer

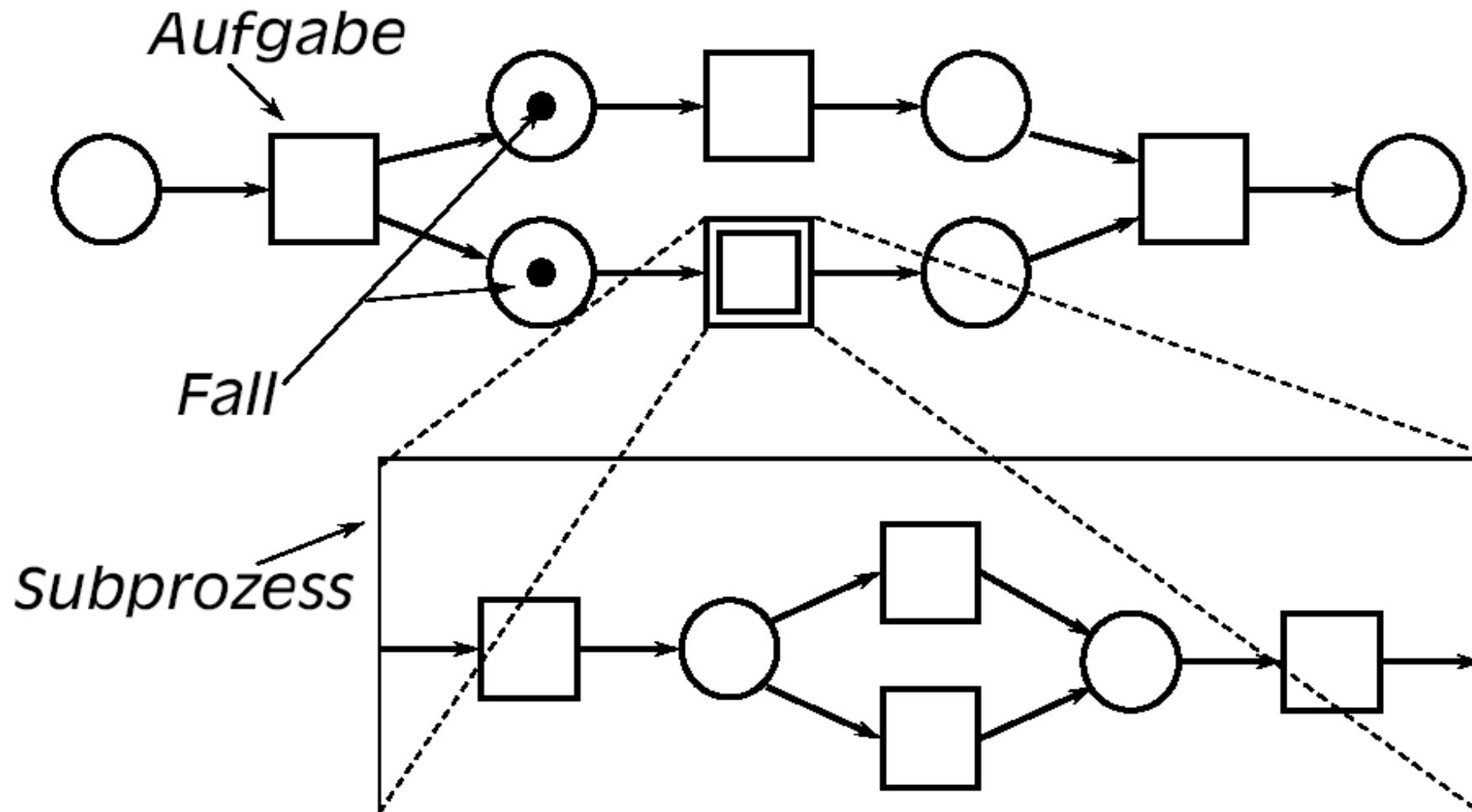


3. Grundregeln der Modellierung mit Petrinetzen

Beispiel: Ein Händler, zwei (individuelle) Käufer



Prozessmodellierung mit Petrinetzen



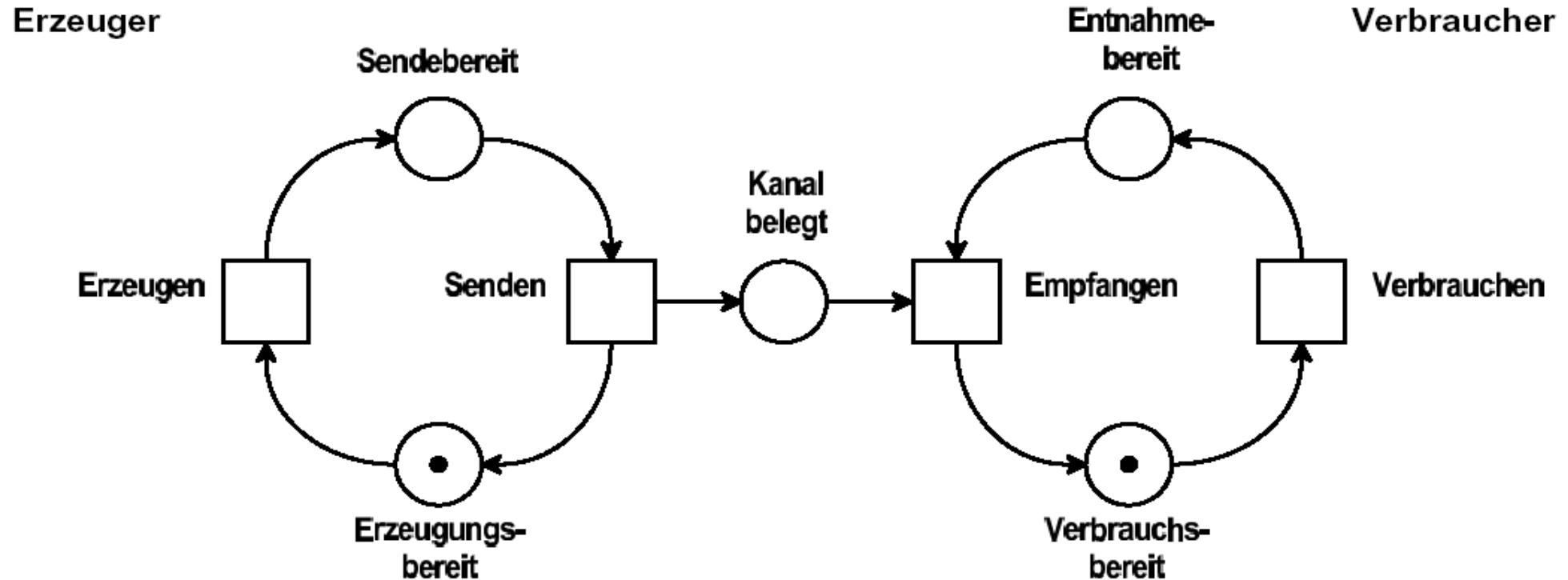
Struktur von Petrinetzen

- Ein **Petrinetz** ist ein Tripel $N = (S, T, F)$ mit
 - S (Stellen), T (Transitionen) sind endliche Mengen,
 - $S \cap T = \emptyset$, (Stellen und Transitionen sind disjunkt),
 - $S \cup T \neq \emptyset$, (Vereinigung von Stellen und Transitionen ist nicht leer),
 - $F \subseteq (S \times T) \cup (T \times S)$ ist eine binäre (Fluss-)Relation über $S \cup T$.
- Alle Stellen und Transitionen eines Netzes heißen **Netzelemente**.

4. Petrinetze formal

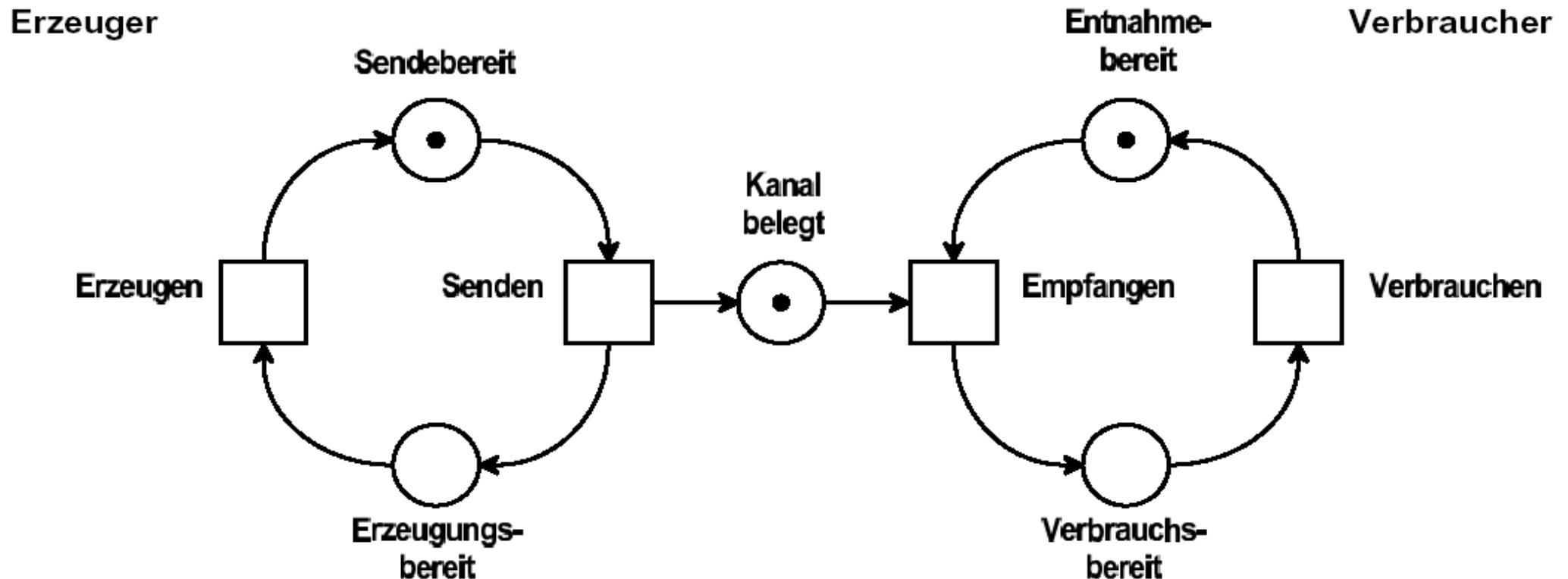


Ein Beispiel (Erzeuger/Verbraucher)



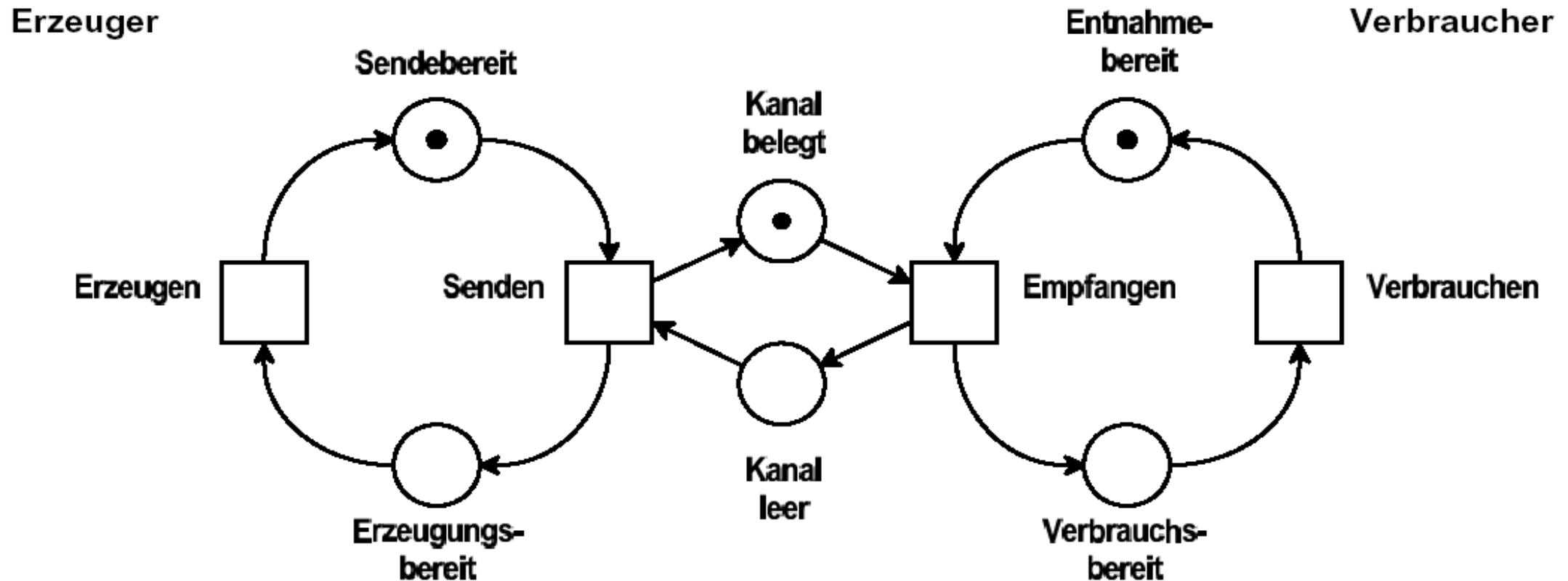
4. Petrinetze formal

Ein Beispiel (Erzeuger/Verbraucher)



4. Petrinetze formal

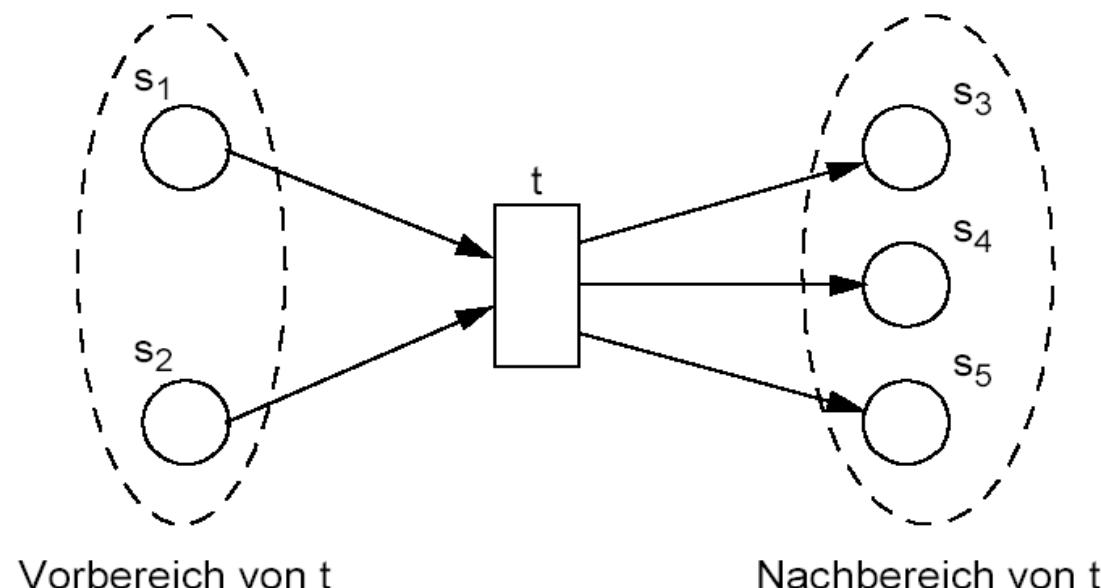
Ein Beispiel (modifizierter Erzeuger/Verbraucher)





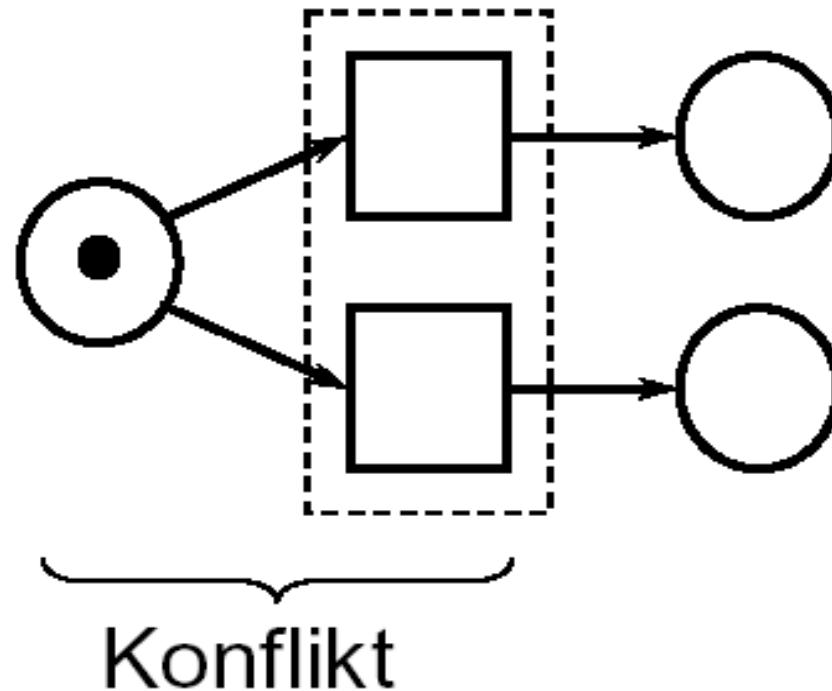
Struktur von Petrinetzen

- **Vorbereich** eines Elements x : Menge aller Eingangs-Knoten (bzw. Input-Knoten) von x , d.h.
 - $x^- = \{ y \mid (y,x) \in F \}$.
- **Nachbereich** eines Elements x : Menge aller Ausgangs-Knoten (bzw. Output-Knoten) von x , d.h.
 - $x^+ = \{ y \mid (x,y) \in F \}$.



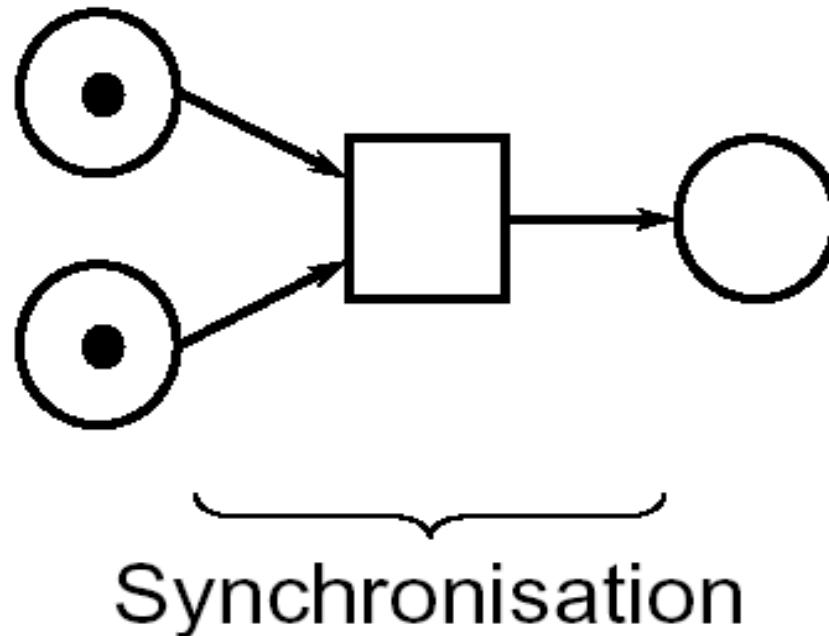
Verzweigungen

- Ein Knoten heißt **vorwärtsverzweigt**, falls $|x \bullet| > 1$.
- Vorwärtsverzweigte Stellen modellieren Alternativen (Konflikte)



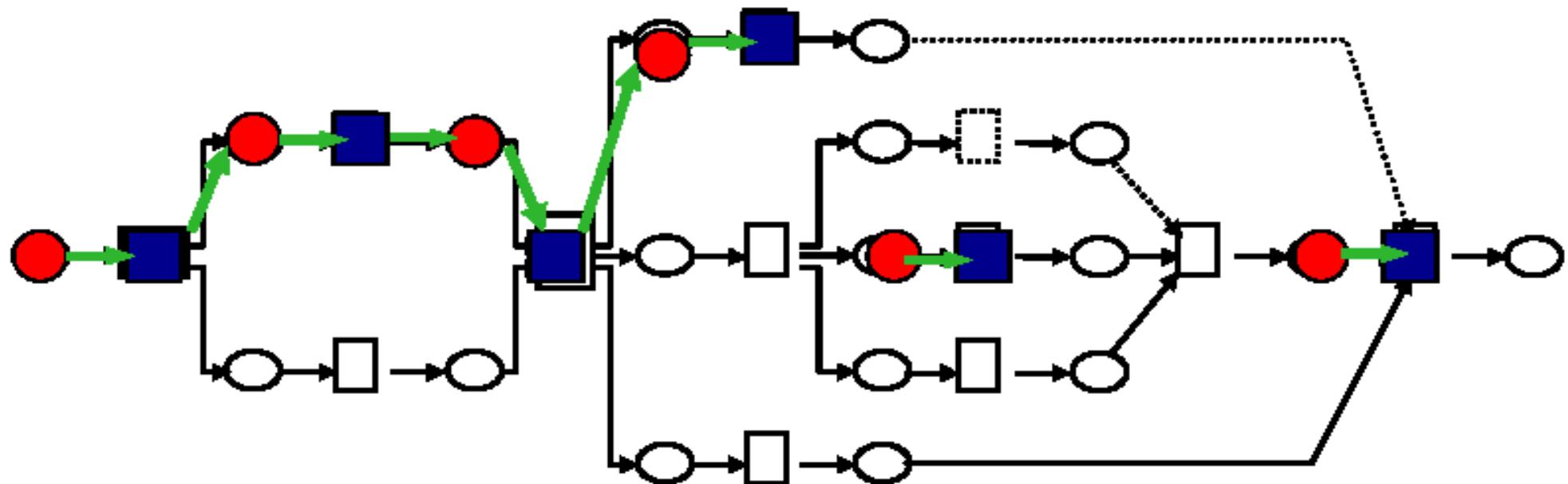
Verzweigungen

- Ein Knoten heißt **rückwärtsverzweigt**, falls $|• x| > 1$.
- Rückwärtsverzweigte Transitionen modellieren Synchronisation.



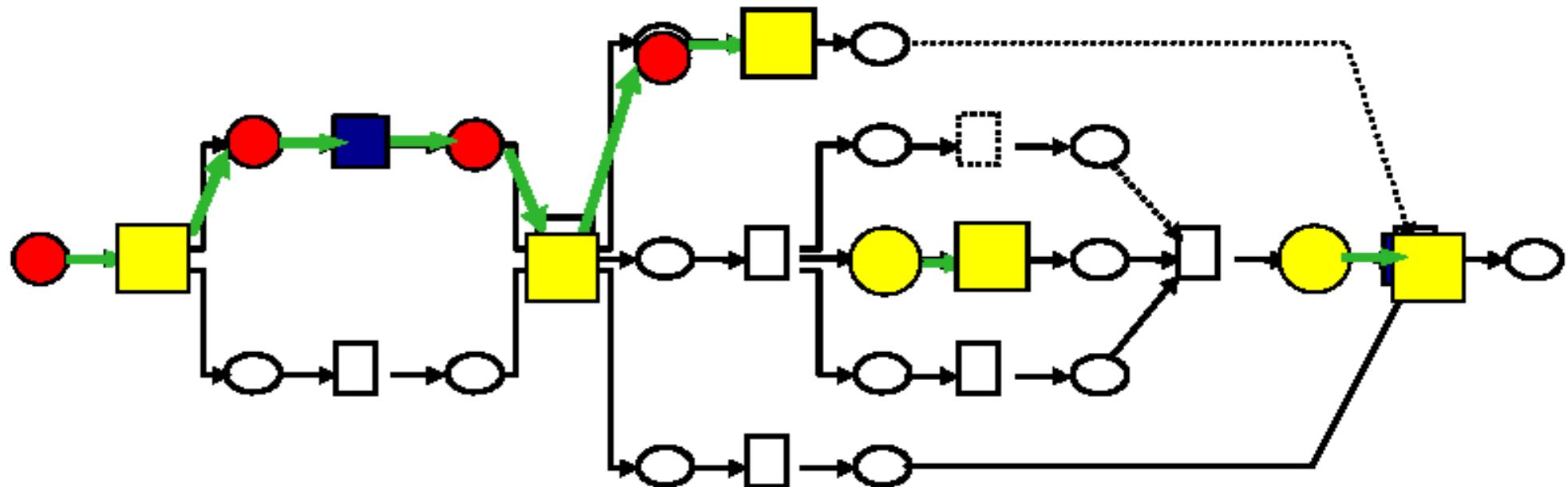
Definition: Teilnetz

- Ein Netz $N' = (S', T', F')$ heißt **Teilnetz** des Netzes $N = (S, T, F)$, wenn
 - $S' \subseteq S$ und
 - $T' \subseteq T$ und
 - $F' = F \cap ((S' \times T') \cup (T' \times S'))$.



Definition: Rand

- Der **Rand** eines Teilnetzes N' (bzgl. des Netzes N) sind diejenigen seiner Knoten, die über Kanten mit dem Restnetz verbunden sind. Er ist also definiert durch: $\text{Rand}(N', N) = \{x \in S' \cup T' \mid (x \bullet \cup \bullet x) \setminus (S' \cup T') \neq \emptyset\}$
(Vor- und Nachbereich von x sind hierbei bzgl. N verstehen)

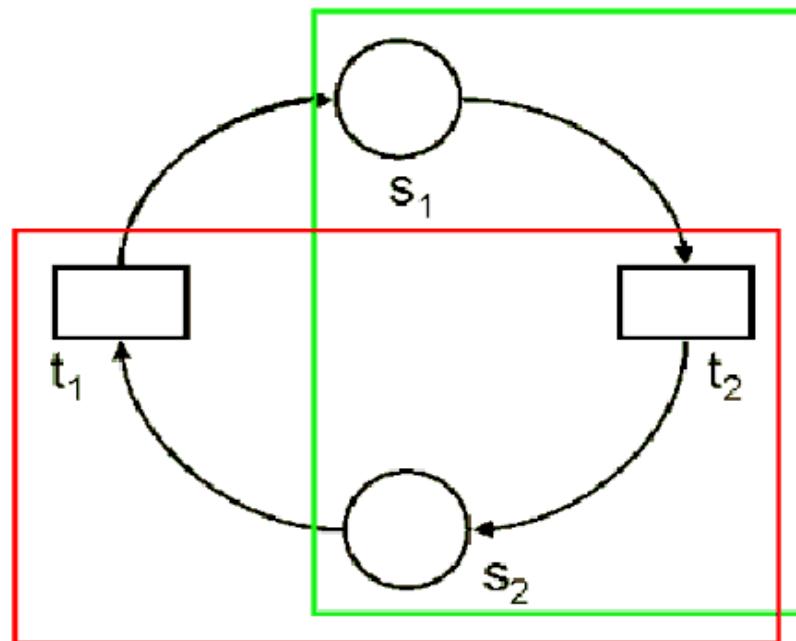




Definition: Rand

- Ein Teil-Netz N' heißt
 - **stellenberandet**, wenn sein Rand nur Stellen enthält, d.h. $\text{Rand}(N', N) \subseteq S'$,
 - **transitionsberandet**, wenn sein Rand nur Transitionen enthält, d.h. $\text{Rand}(N', N) \subseteq T'$.
- Ein stellenberandetes Teilnetz lässt sich durch eine Stelle, ein transitionsberandetes durch eine Transition ersetzen.

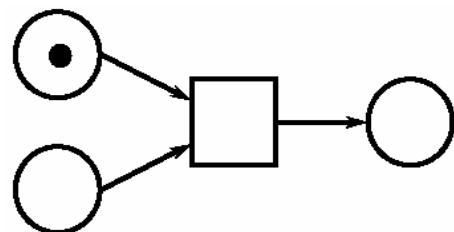
- Beispiel:
 - **stellenberandet**
 - **transitionsberandet**



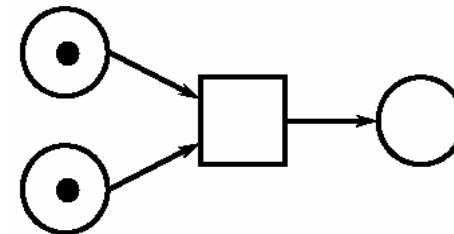
Definition: Markierung

- Die Belegung der Stellen heißt **Markierung** und ist der Zustand des Petri-Netzes.
- Eine **Markierung** m eines Netzes $N=(S,T,F)$ ist eine Abbildung $m: S \rightarrow \mathbb{N}$.
- Eine Markierung m **aktiviert** eine Transition $t \in T$, wenn $m(s) > 0$ für alle $s \in \bullet t$.
- Kontroll-Frage: In welchem Fall ist die Transition aktiviert?

A.)



B.)



Definition: Markiertes Petrinetz

- Ein Netz N mit Markierung m ist ein **markiertes Petrinetz**.
- Ein markiertes Petrinetz ist also ein 4-Tupel (S, T, F, m) .
- Falls $t \in N$ unter m aktiviert ist, kann t **schalten**. Dies führt zu einer **Folgemarkierung m'** , definiert durch

$$m'(s) = \begin{cases} m(s), & \text{falls } s \notin \bullet t \text{ und } s \notin t \bullet \\ m(s)-1, & \text{falls } s \in \bullet t \text{ und } s \notin t \bullet \\ m(s)+1, & \text{falls } s \notin \bullet t \text{ und } s \in t \bullet \\ m(s), & \text{falls } s \in \bullet t \text{ und } s \in t \bullet \end{cases}$$

Schreibweise: $m \xrightarrow{t} m'$.

Definition: Schaltfolge

- Sei m eine Markierung eines Petrinetzes N . Falls $m \xrightarrow{t_1} m_1, m_1 \xrightarrow{t_2} m_2, \dots, m_{n-1} \xrightarrow{t_n} m_n$ Schaltvorgänge sind, ist $SF = t_1, t_2, \dots, t_n$ eine Schaltfolge von m nach m_n :
 $(m \xrightarrow{SF} m_n)$
- Die leere Sequenz ε ist die Schaltfolge $\varepsilon : m \rightarrow m$ für jede Markierung m .

Definition: Erreichbarkeit

- Wir schreiben $m \rightarrow^* m'$ und nennen m' von m **erreichbar**, wenn $m \xrightarrow{SF} m'$ für irgendeine Schaltfolge SF gilt.
- $[m]$ bezeichnet die Menge aller von m erreichbaren **Markierungen**.
- Das Verhalten eines markierten Petrinetzes wird beschrieben durch die Menge seiner Schaltfolgen.
- Eine kompaktere Repräsentation liefert der Markierungsgraph.
- Eine **Markierung** eines Petrinetzes heißt **erreichbar**, falls es eine Schaltfolge der Transitionen gibt, welche die Startmarkierung in diese Markierung überführt.

Geschäftsprozess-Management

Prof. Dr.-Ing. Andreas Ittner

Email: ittner@hs-mittweida.de

WWW: www.hs-mittweida.de/~ittner

Tel.: +49(0)3727-58-1288

Mob.: +49(0)177-5555-347

Gliederung (vorläufig)

- Motivation
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs, BPMN, ...
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards

Gliederung:

1. Einführung in die Modellierung,
2. Geschäftsprozess-Modellierung
3. Grundregeln der Modellierung mit Petrinetzen,
4. Petrinetze formal,
5. High-level Petrinetze,
6. Grundregeln der EPK-Modellierung,
7. Verknüpfungsoperatoren bei EPK,
8. Erweiterte EPK und ARIS,
9. EPK vs. Petrinetze,
10. BPMN.

Beim praktischen Einsatz von „klassischen“ Petrinetzen treten oft Probleme auf:

- Die Modelle werden zu groß und komplex.
- Die Modellierung ist langwierig und kompliziert.
- Zeit, Kosten und Daten können nicht modelliert werden.

High-Level Petrinetze sind Petrinetze mit den folgenden Erweiterungen:

- **Trigger**
 - zur Modellierung äußerer Einflüsse (Wartezeiten u.ä.)
- **Unterscheidbare Marken**
 - zur Modellierung von Attributen
 - auch: gefärbte Marken, colored tokens
- **Zeit**
 - zur Performance-Analyse
- **Hierarchie**
 - zur Strukturierung der Modelle
 - Modellierung auf verschiedenen Abstraktionsebenen
- **Beschriftung mit Funktion**
- U.a.

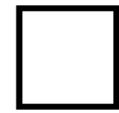
Trigger:

- Die zu modellierenden Prozesse sind oft von äußeren Einflussfaktoren abhängig,
- Zum Beispiel muss bei bestimmten Aufgaben
 - die Ankunft einer Nachricht,
 - die Ankunft einer Akte,
 - ein Anruf zur Auftragsbestätigung,
 - der Arbeitsbeginn einer Ressource,
 - o.ä.
- abgewartet werden.
- Einige Aufgabe erfordern deshalb Trigger.

Trigger: Arten von Aufgaben

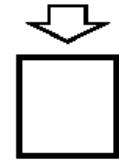
- **Automatisch**

Kein Trigger erforderlich.



- **Benutzer**

Initiiert durch eine Ressource.



- **Extern**

Ein externer Event (z.B. Nachricht, Anruf) ist erforderlich.



- **Zeit**

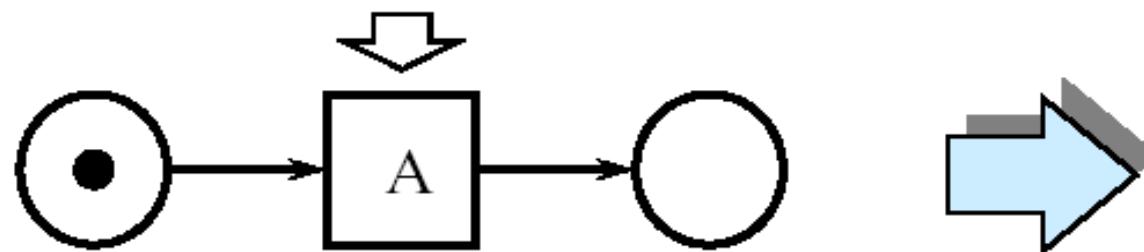
Zeitdauern oder Zeitpunkte werden berücksichtigt.



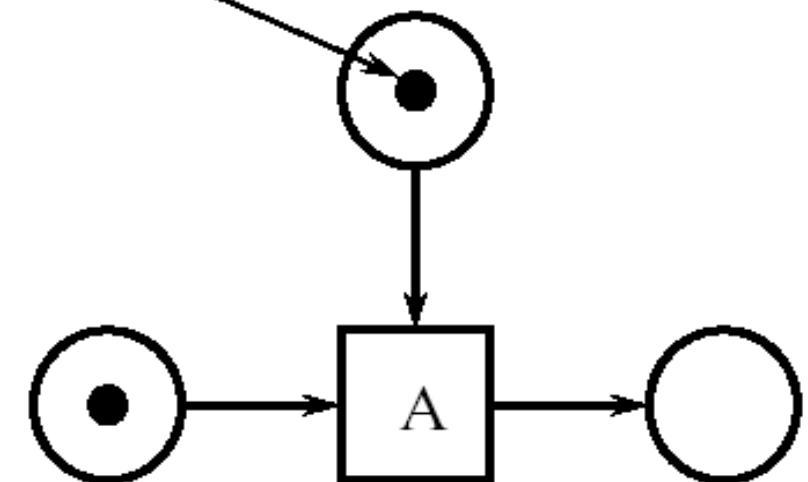
5 High-Level Petrinetze

Trigger:

Aufgabe benötigt Trigger

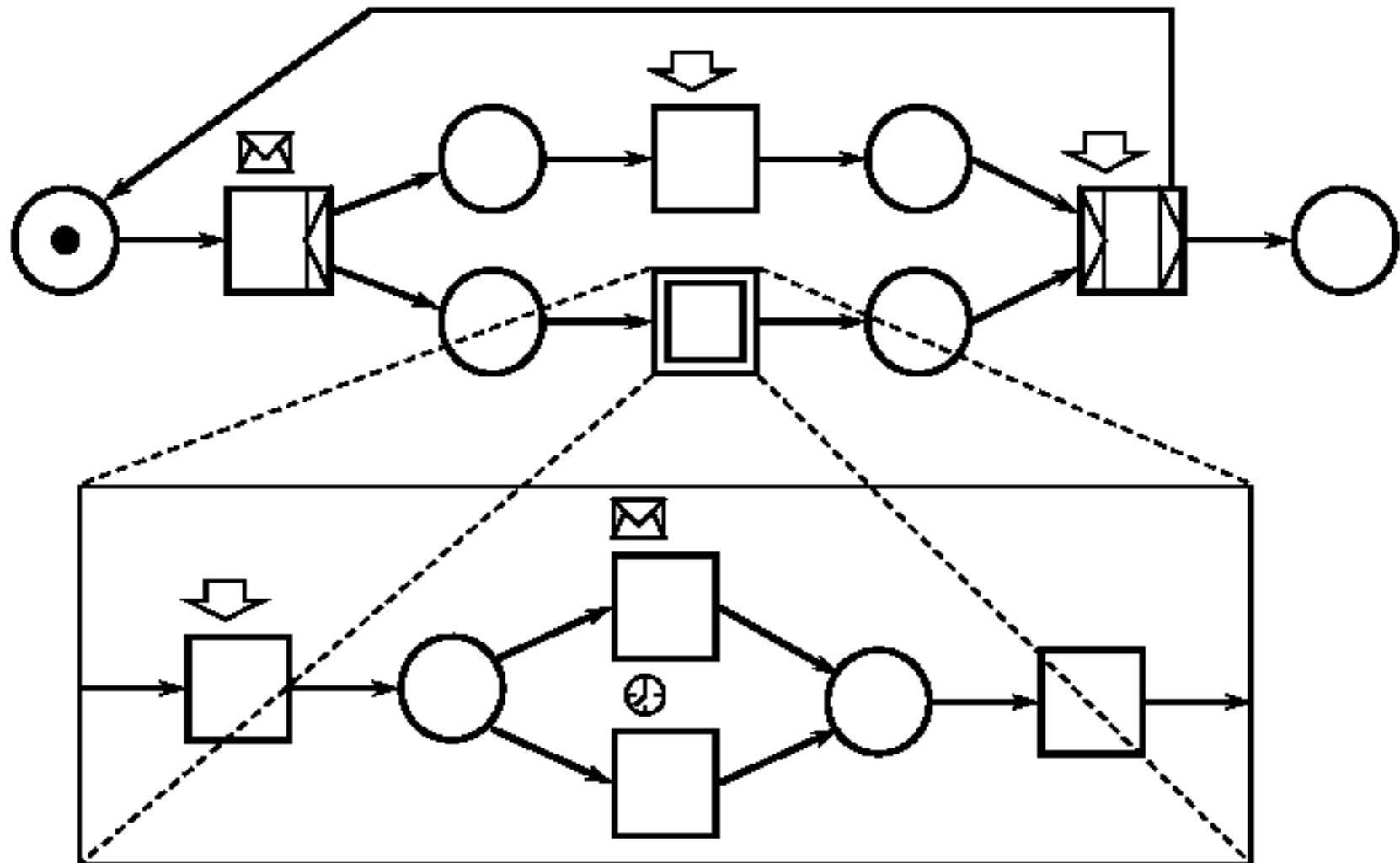


Trigger-Marke



5 High-Level Petrinetze

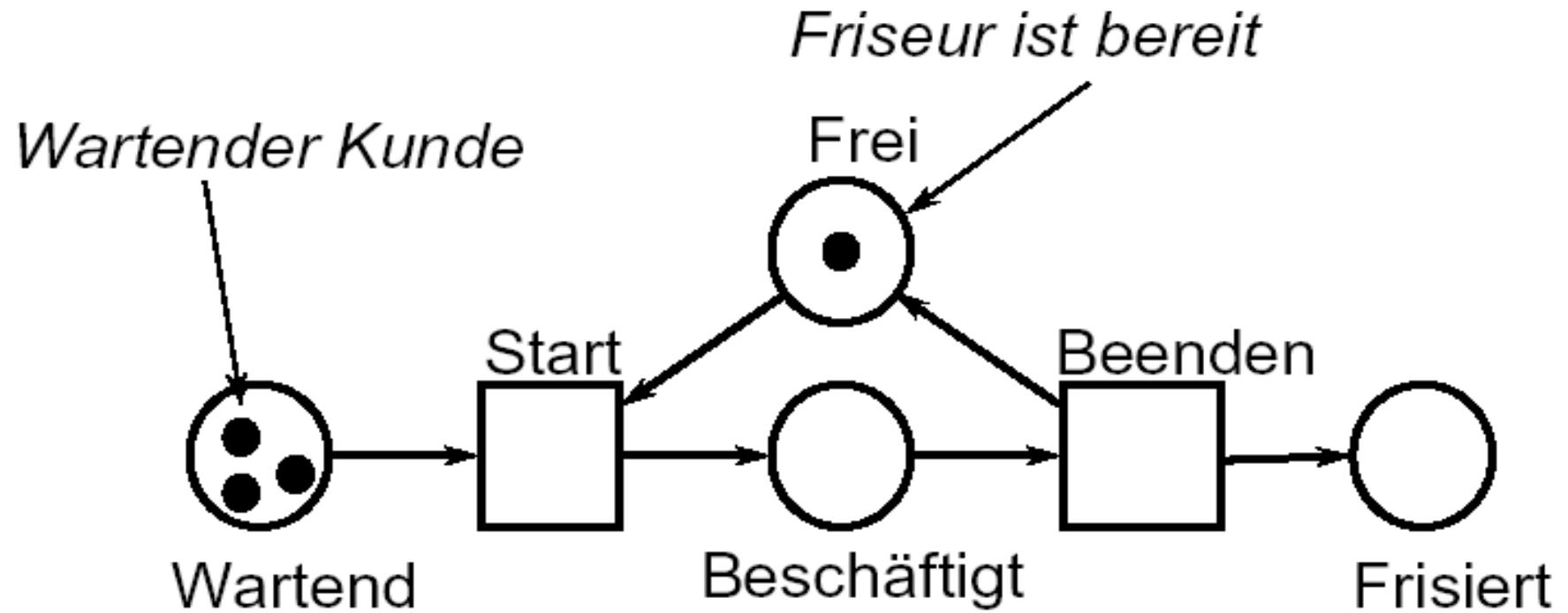
Trigger:



5 High-Level Petrinetze

Unterscheidbare Marken:

Beispiel: beim Friseur (I)



Anmerkung: man bemerke die problemlose Modellierung einer Situation mit mehreren Friseuren

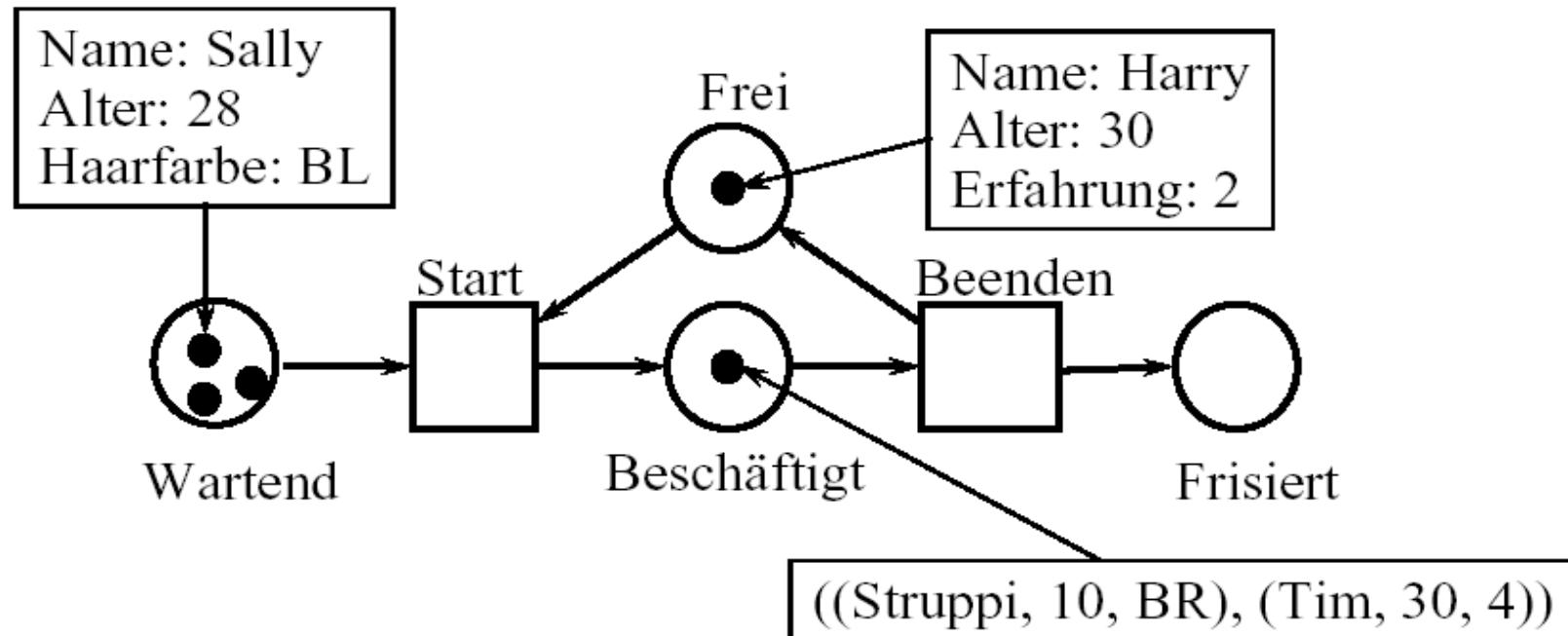
5 High-Level Petrinetze

Unterscheidbare Marken:

Beispiel: beim Friseur (II)

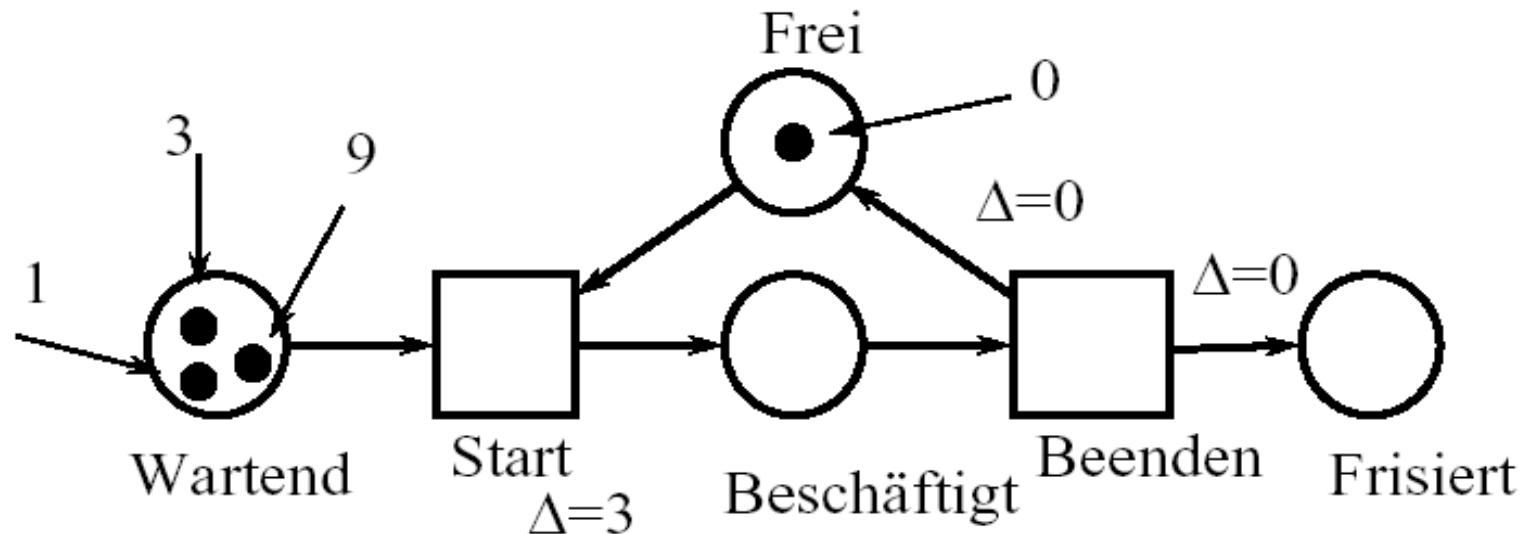
Eine unterscheidbare Marke

- stellt ein Objekt mit einer Menge von Attributen dar.
- beinhaltet Werte für alle Attribute.



Die Erweiterung mit Zeit:

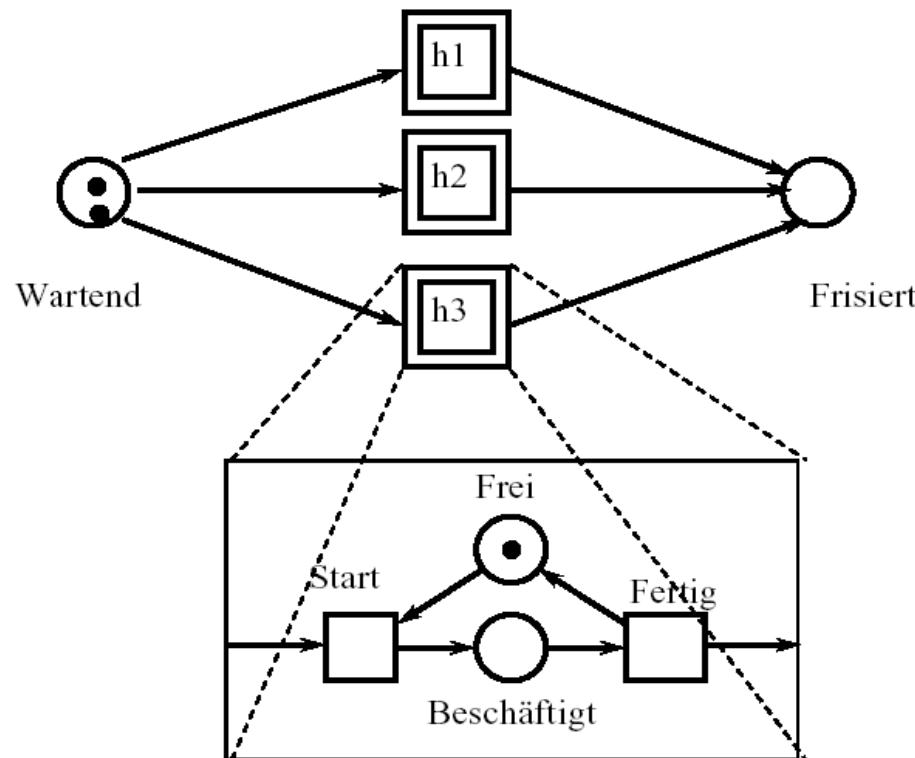
- Für Performance-Analysen werden Zeitdauern, Verzögerungen etc. benötigt.
- Jede Marke bekommt einen Zeitstempel.
- Durch Transitionen wird das Alter (und damit die Verfügbarkeit) der Marke neu gesetzt.



- verschiedene Zeitkonzepte (hier: Bestimmung der Verzögerung jedes Tokens)

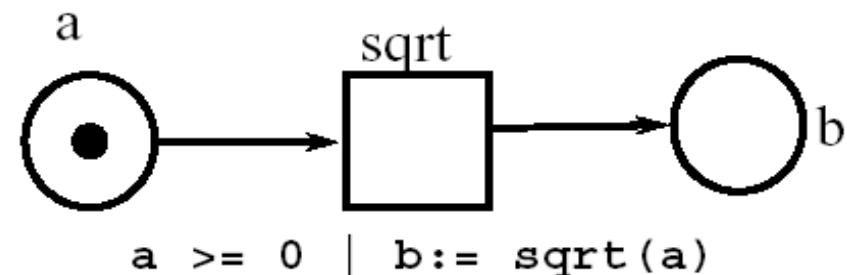
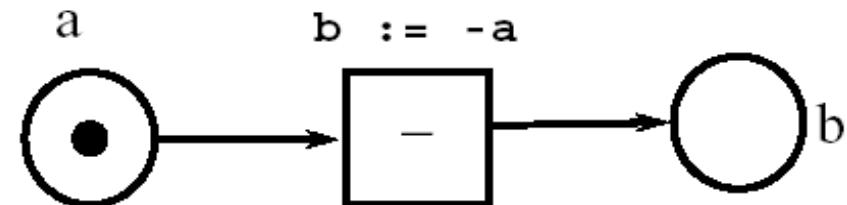
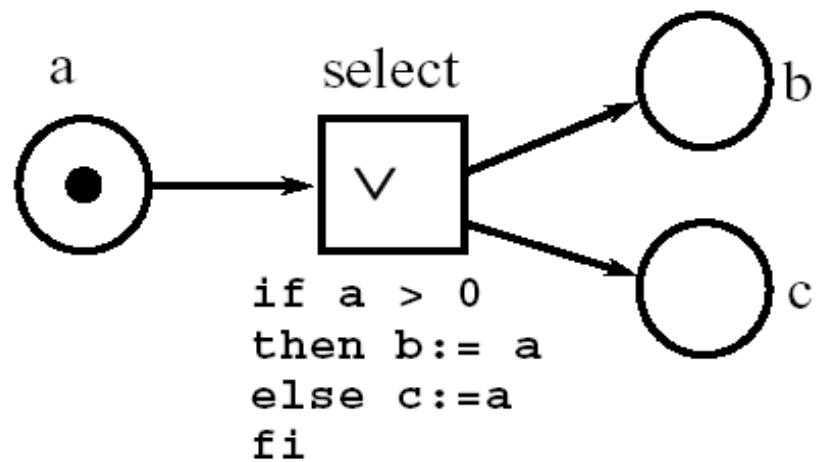
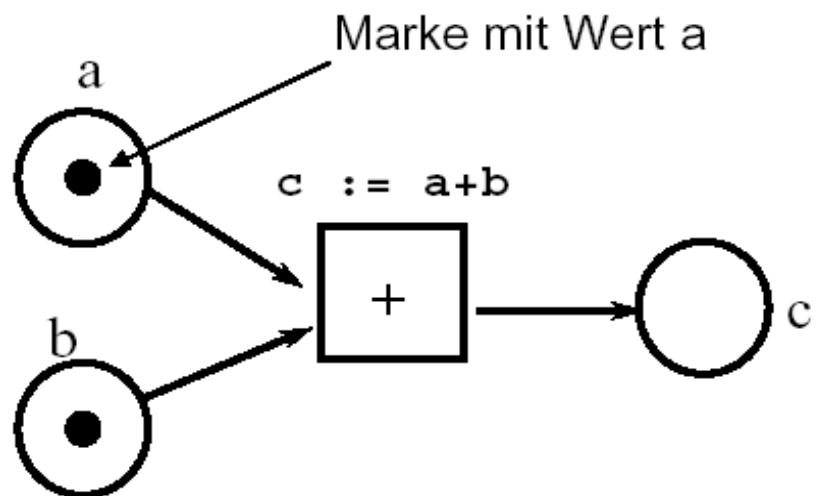
Die Erweiterung um Hierarchie-Konzepte:

- Ein Mechanismus zur Strukturierung komplexer Modelle ermöglicht das Modellieren und die Darstellung auf verschiedenen Abstraktionsstufen
- Ein Netzelement wird durch ein entsprechend benanntes Teilnetz ersetzt.



5 High-Level Petrinetze

Beschriftung mit Funktion:



5 High-Level Petrinetze

Weiterhin kann für jede Transition spezifiziert werden:

- der Name,
- die Anzahl der produzierten Marken,
- die Anzahl der konsumierten Marken,
- die Werte der entsprechenden Attribute,
- (optional) eine Schaltbedingung (auch: guard),
- ...

Gliederung (vorläufig)

- Motivation
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs, BPMN, ...
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards

Gliederung:

1. Einführung in die Modellierung,
2. Geschäftsprozess-Modellierung
3. Grundregeln der Modellierung mit Petrinetzen,
4. Petrinetze formal,
5. High-level Petrinetze,
6. Grundregeln der EPK-Modellierung,
7. Verknüpfungsoperatoren bei EPK,
8. Erweiterte EPK und ARIS,
9. EPK vs. Petrinetze,
10. BPMN.

6 Grundregeln der EPK-Modellierung

Die EPK-Methode (EPK: Ereignisgesteuerte Prozesskette)

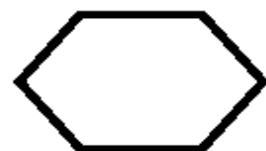
- graphische Modellierungssprache,
- basiert auf Petrinetztheorie,
- kann verstanden werden als eine Variante, die um logische Verknüpfungsoperatoren erweitert wurde,
- ist semiformal, d.h. enthält sowohl textuelle als auch graphische Modellierungselemente,
- 1992 entwickelt von Prof. Scheer (IDS Prof. Scheer GmbH) und Mitarbeitern,
- hoher Verbreitungsgrad in Deutschland: ARIS-Toolset bzw. SAP R/3-Analyzer.

6 Grundregeln der EPK-Modellierung

EPK ist ein gerichteter Graph und besteht aus den Elementen:

Knoten des Graphen:

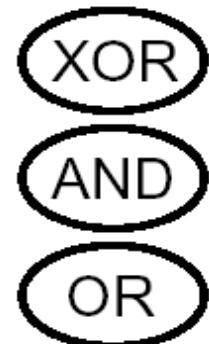
Ereignis



Funktion



Verknüpfungs-
operatoren



Kanten des Graphen:

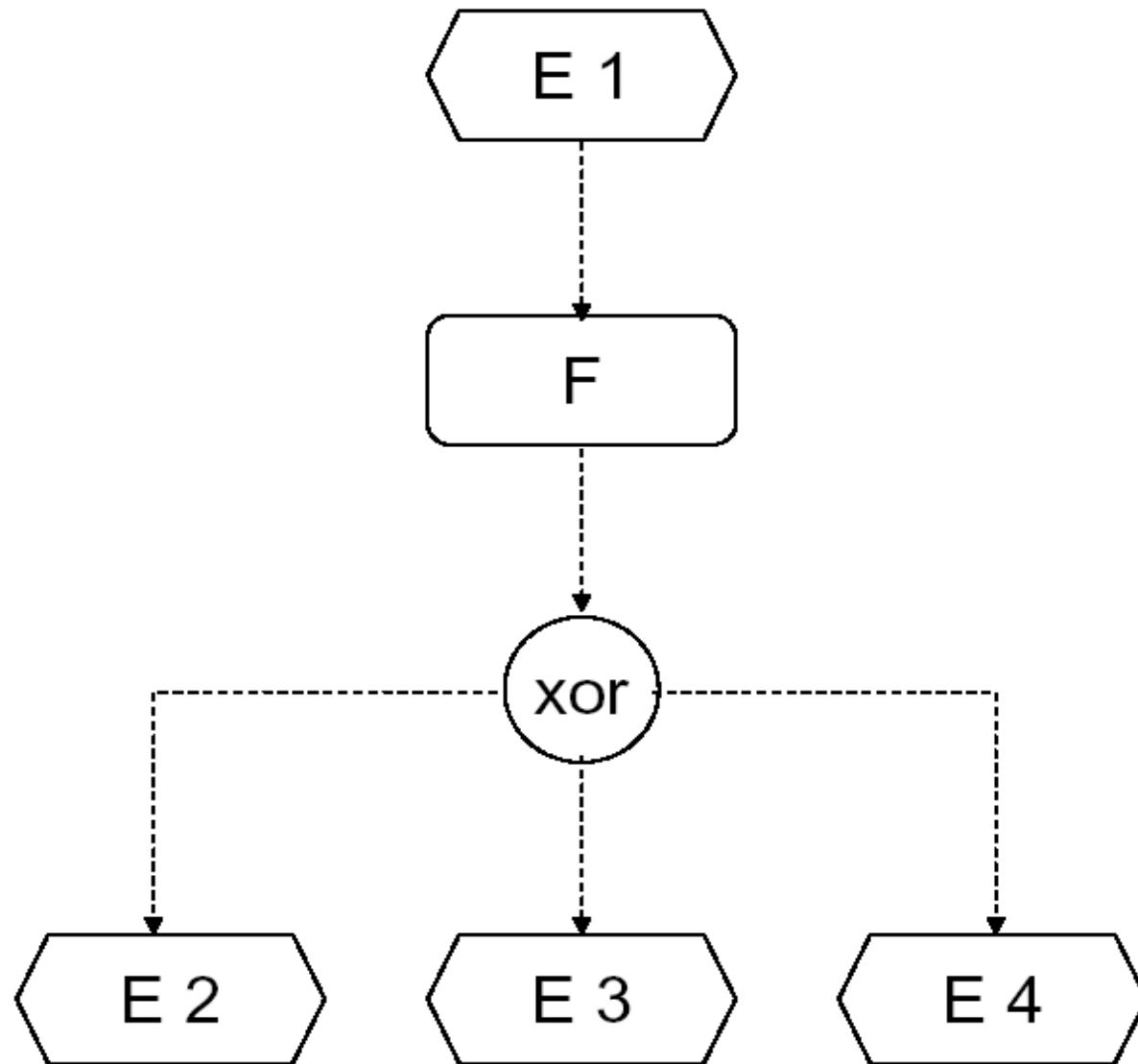
Abhängigkeit zwischen **Ereignis**
und **Funktion**



(an geeigneten Stellen ○
zwischengeschaltet)



6 Grundregeln der EPK-Modellierung



6 Grundregeln der EPK-Modellierung

In ihrem Grundprinzip besteht die EPK aus

- aktiven Komponenten (**Funktionen**), die etwas durchführen,
- passiven Komponenten (**Ereignissen**), die Aktivitäten auslösen.

Ereignisse sind

- Auslöser von Funktionen und
- deren Ergebnis.

Ereignisse repräsentieren einen eingetretenen **betriebswirtschaftlichen Zustand**.

6 Grundregeln der EPK-Modellierung

Ereignisse:

Syntaktische Regel für „Ereignis“:

- Einem vorangestellten **Substantiv** folgt immer das **Partizip Perfekt** des gewählten Verbs.

Beispiele:

- Kundenauftrag *ist eingetroffen*.
- Materialstamm *ist angelegt*.
- Rechnung *ist gebucht*.

6 Grundregeln der EPK-Modellierung

Funktionen:

Funktionen transformieren Input- in Outputdaten
und:

Eine Funktion hat die ***Entscheidungskompetenz*** über den weiteren Ablauf
(Ereignisse haben keine solche Entscheidungskompetenz)

Funktionen können soweit unterteilt werden, bis sie einen ***betriebswirtschaftlich nicht weiter sinnvoll unterteilbaren*** Vorgang darstellen:

- z.B.: Auftragsbearbeitung – Annahme Telefonanruf | ...

6 Grundregeln der EPK-Modellierung

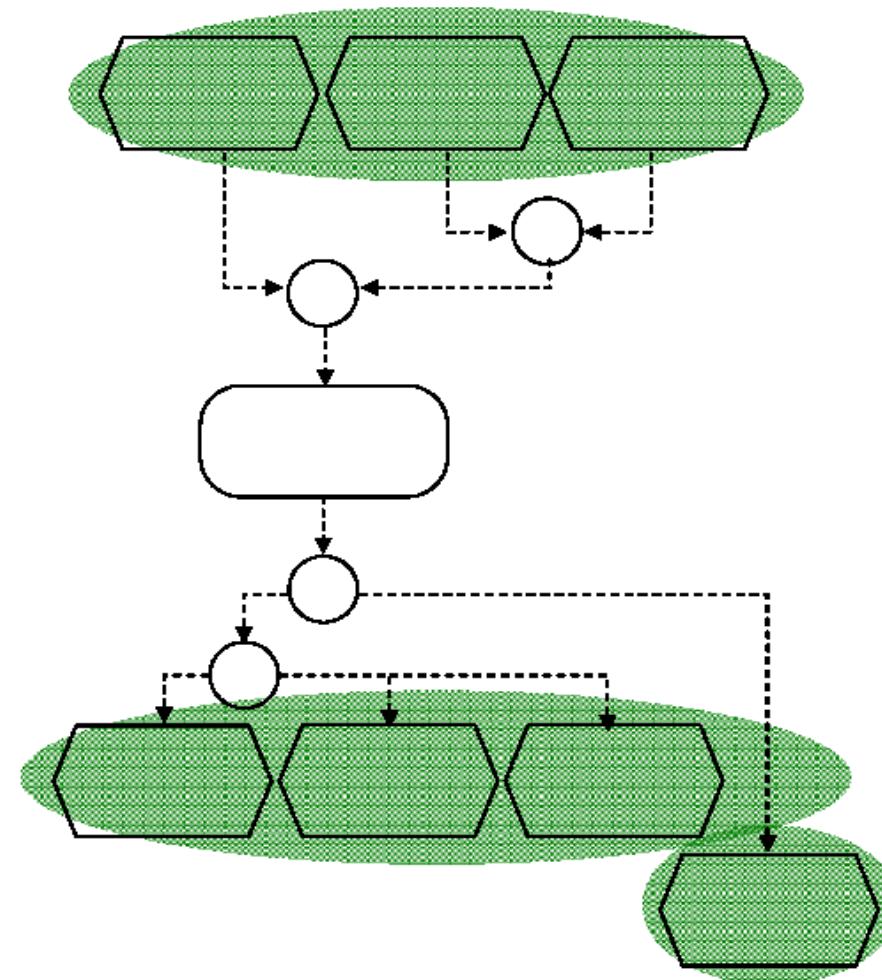
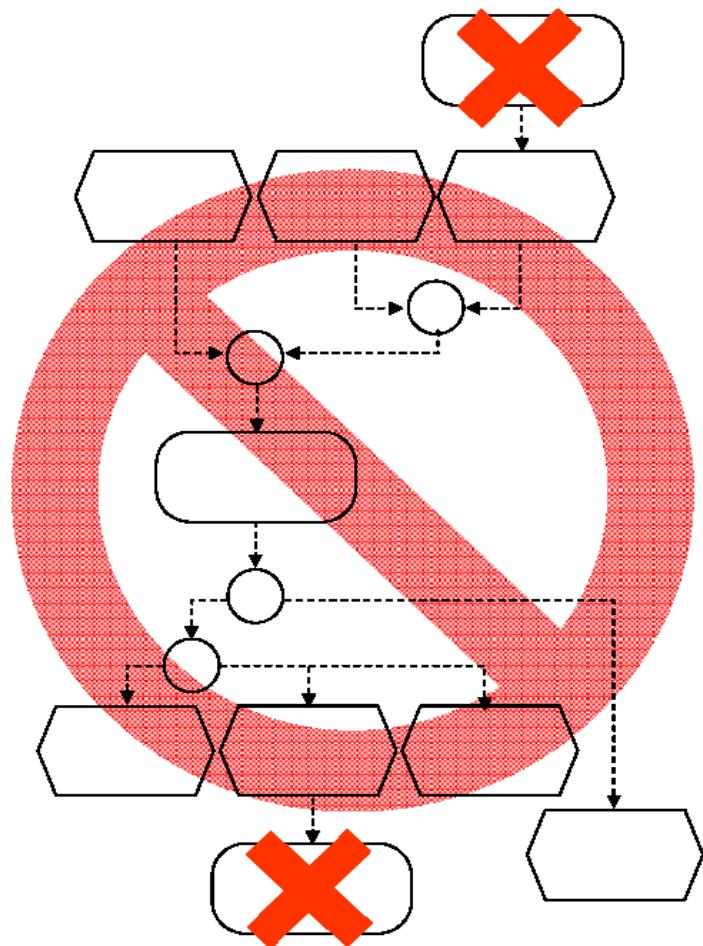
Verknüpfung:

Die beiden Grundelemente Ereignis - Funktion werden **direkt** oder über verschiedene **Verknüpfungsoperatoren** verbunden.

Durch die Zuordnung von Ereignissen zu Funktionen, die wiederum ein oder mehrere Ereignisse erzeugen können, erhält man einen **zusammenhängenden Aufgaben- oder Funktionsablauf**.

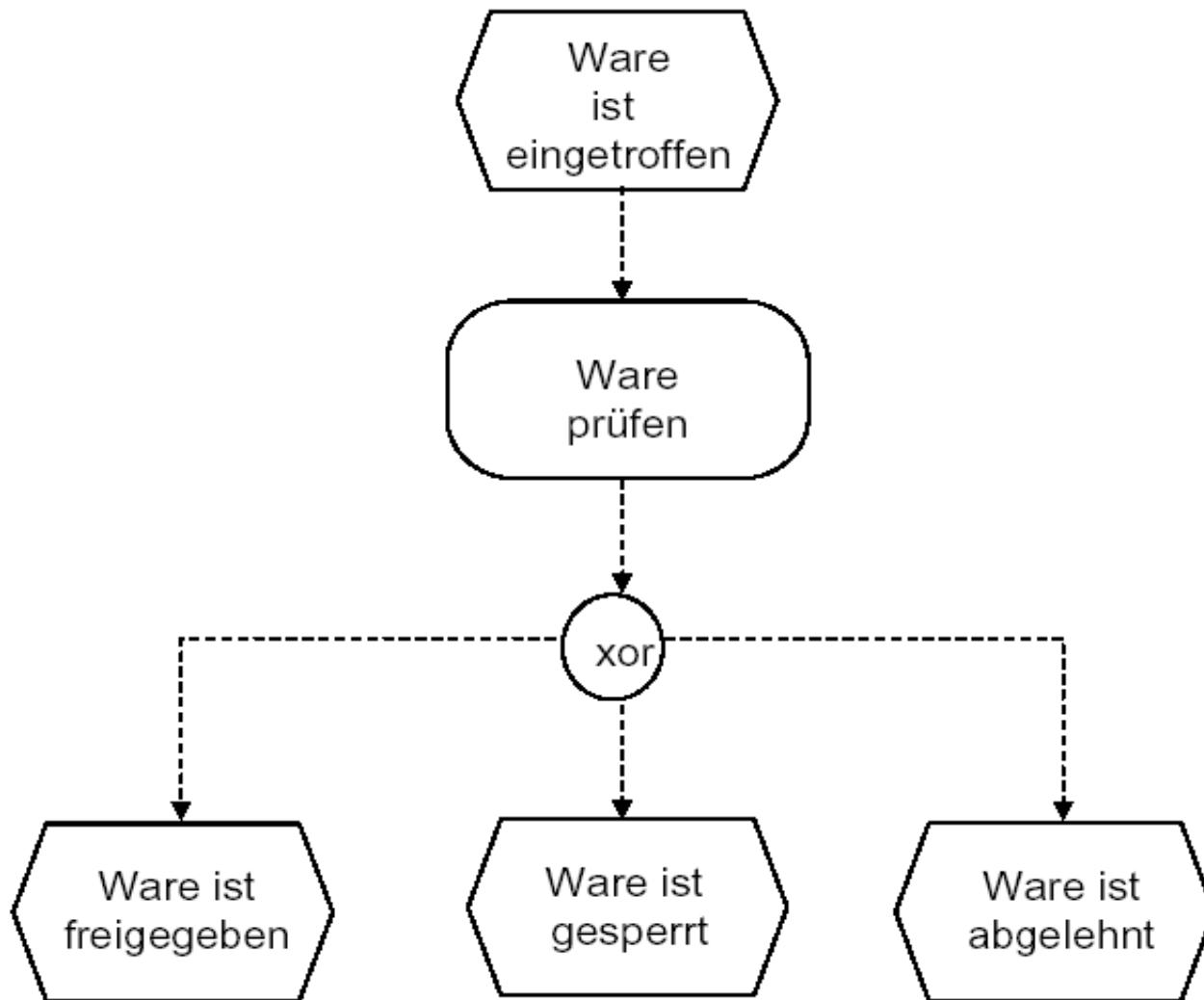
6 Grundregeln der EPK-Modellierung

Jede EPK beginnt mit einem **Startereignis** (oder mehreren) und wird mit einem **Endereignis** (oder mehreren) abgeschlossen.



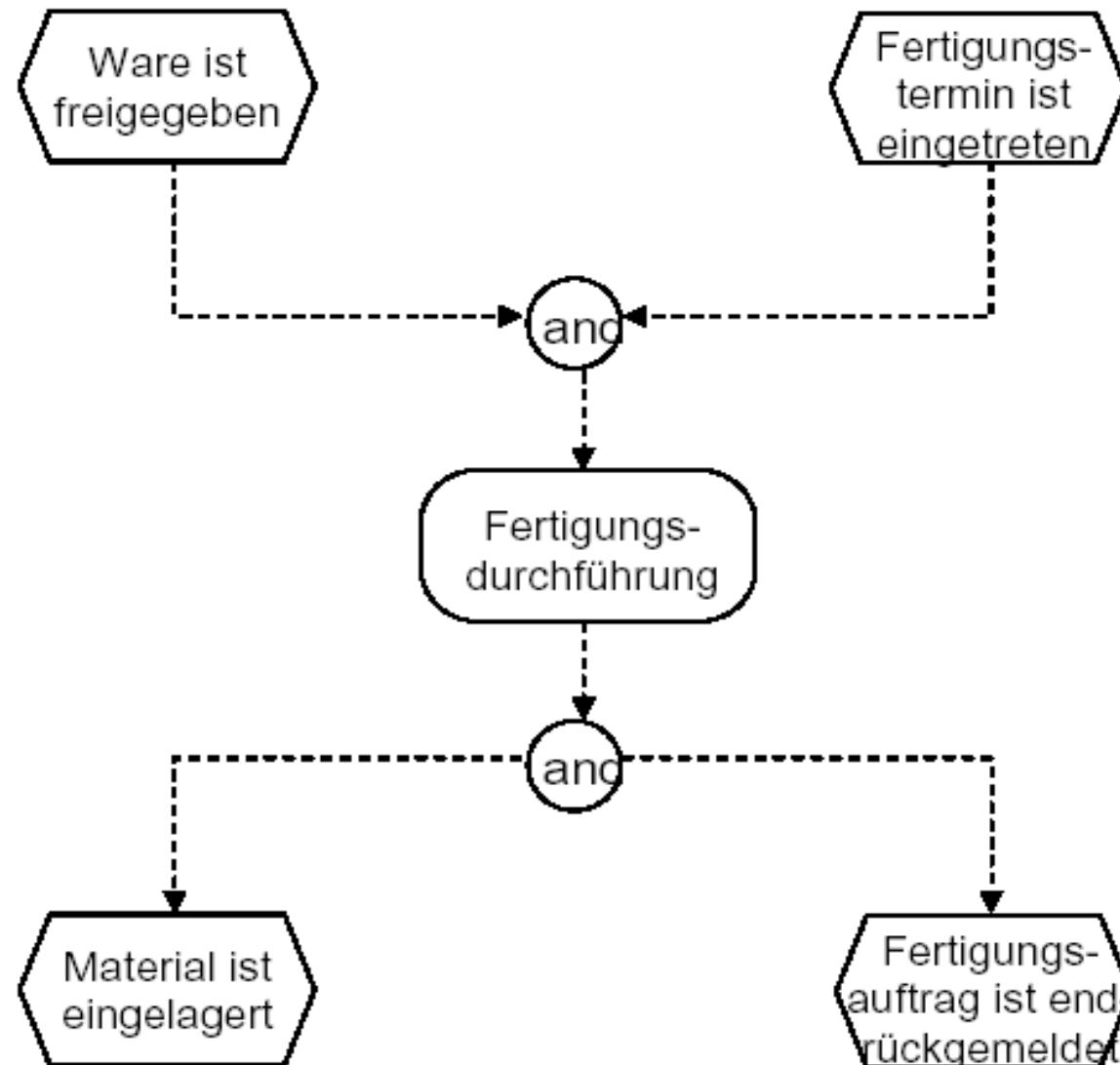
6 Grundregeln der EPK-Modellierung

Beispiel 1: Wareneingangsbearbeitung



6 Grundregeln der EPK-Modellierung

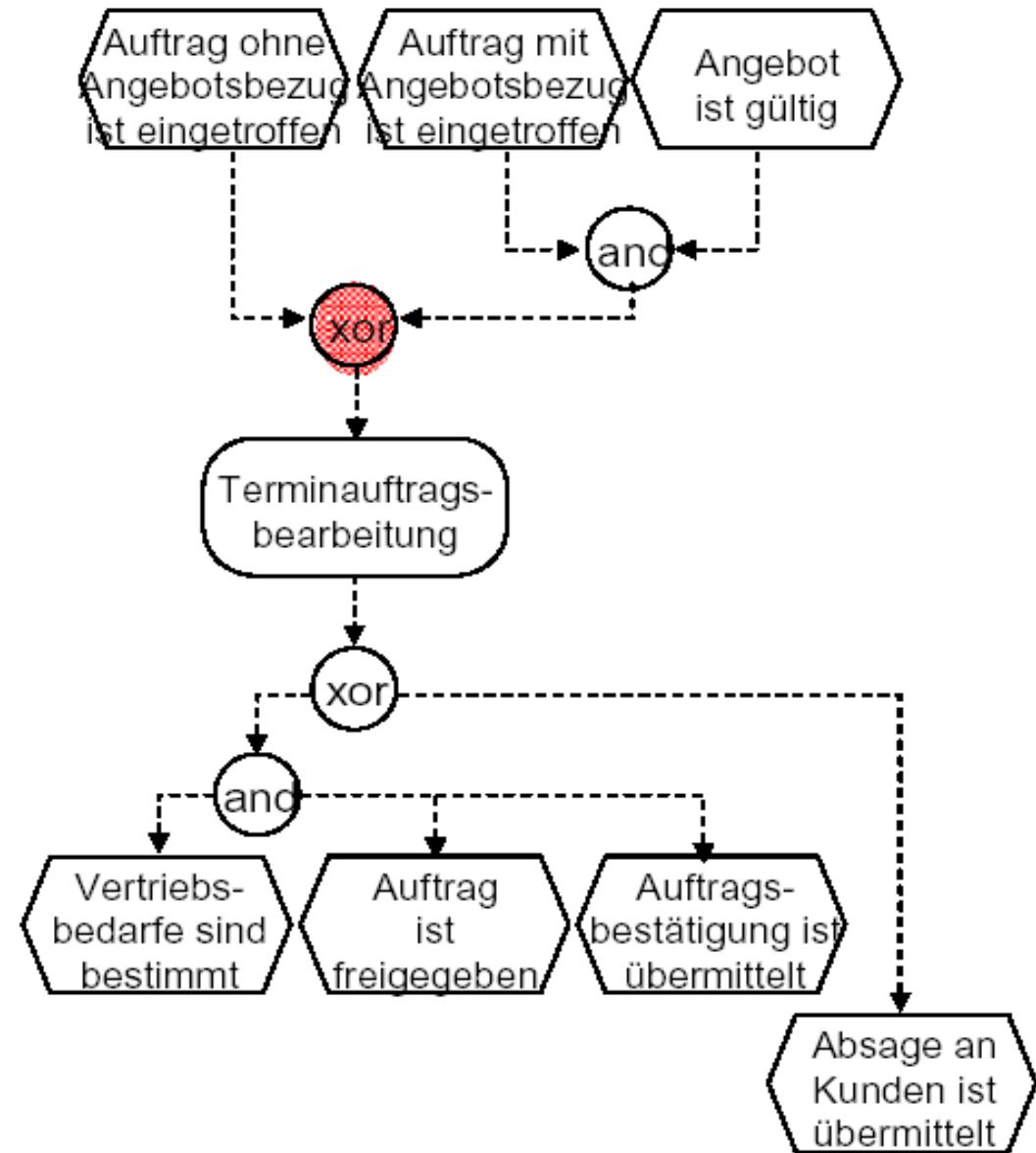
Beispiel 2: Fertigungsdurchführung



6 Grundregeln der EPK-Modellierung

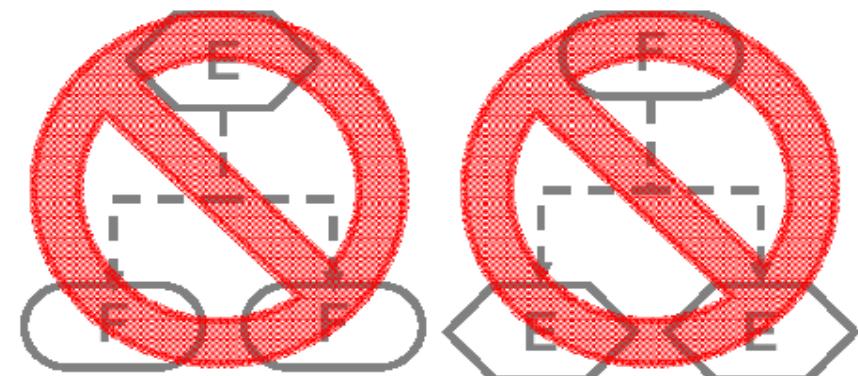
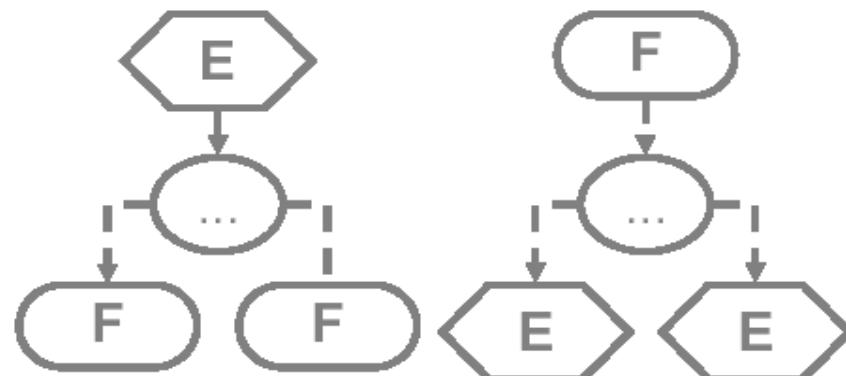
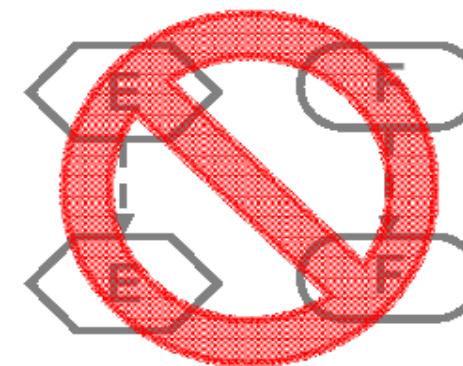
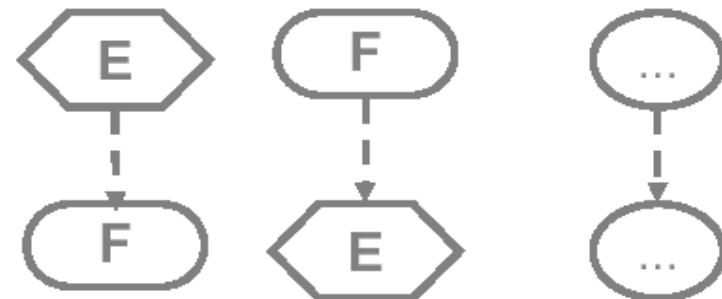
Beispiel 3: Auftragsbearbeitung

Es kann vernünftigerweise
nur **eines** der beiden Ereignisse
eingetreten sein!



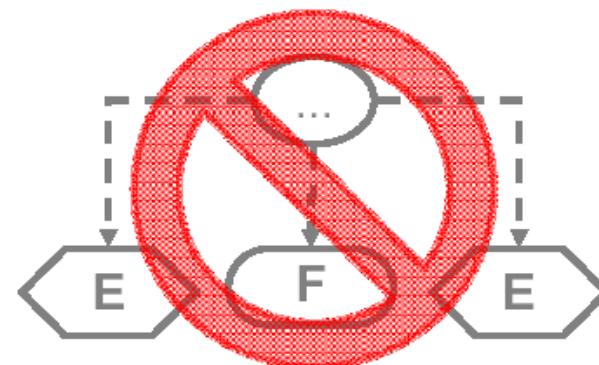
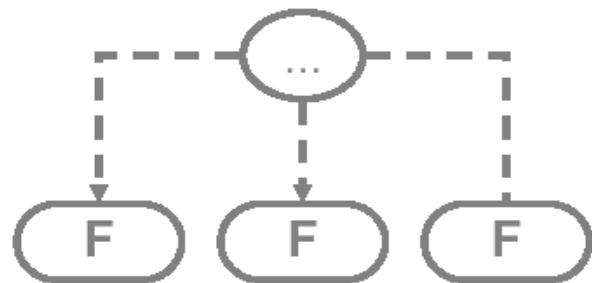
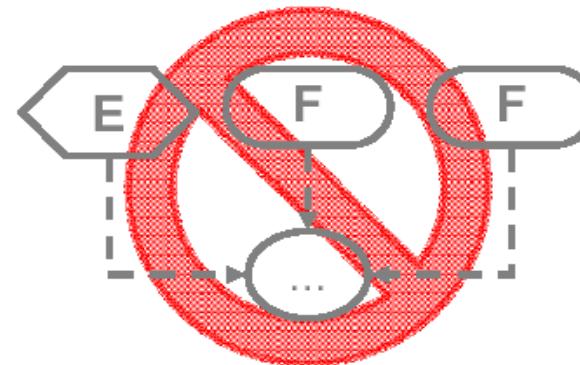
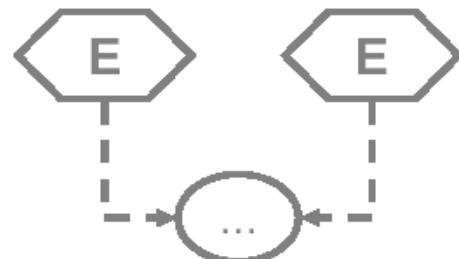
7 Verknüpfungsoperatoren bei EPK

- Jede Kante verbindet in der Regel zwei Knoten von jeweils unterschiedlichem Typ.
- Nur die Verknüpfungsoperatoren verzweigen, sie verbinden Ereignisse mit Funktionen u.u. (sie können auch miteinander verbunden werden).



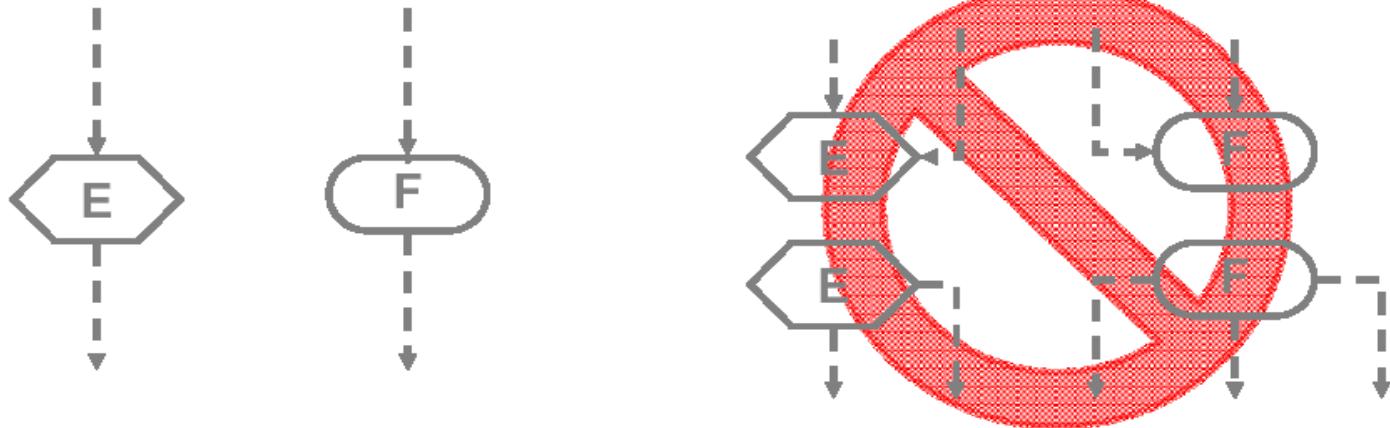
7 Verknüpfungsoperatoren bei EPK

- Die Eingänge eines Verknüpfungsoperators sind entweder **alle vom Typ Ereignis** oder **alle vom Typ Funktion**, ebenso sind seine Ausgänge.

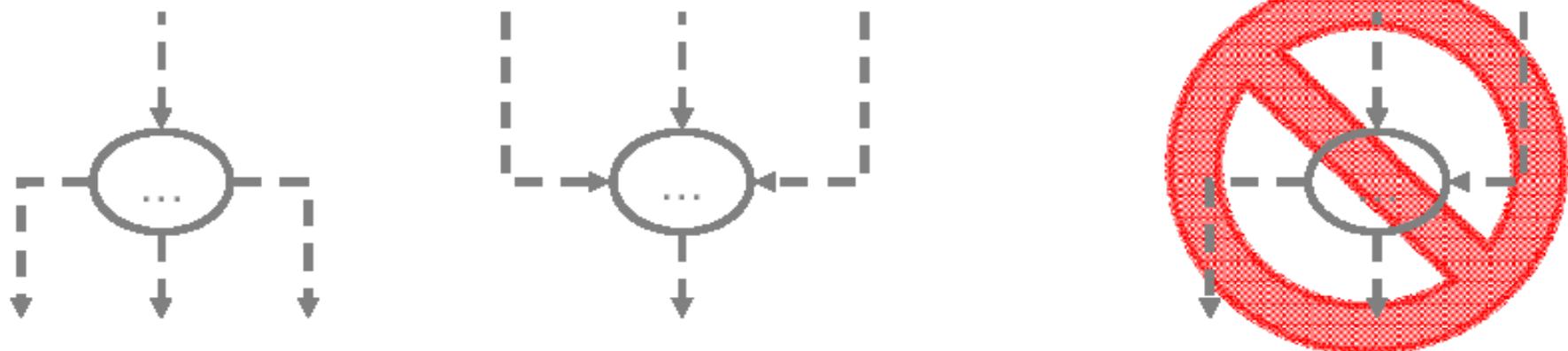


7 Verknüpfungsoperatoren bei EPK

- Ereignisse und Funktionen dürfen nur einen Eingang und einen Ausgang haben



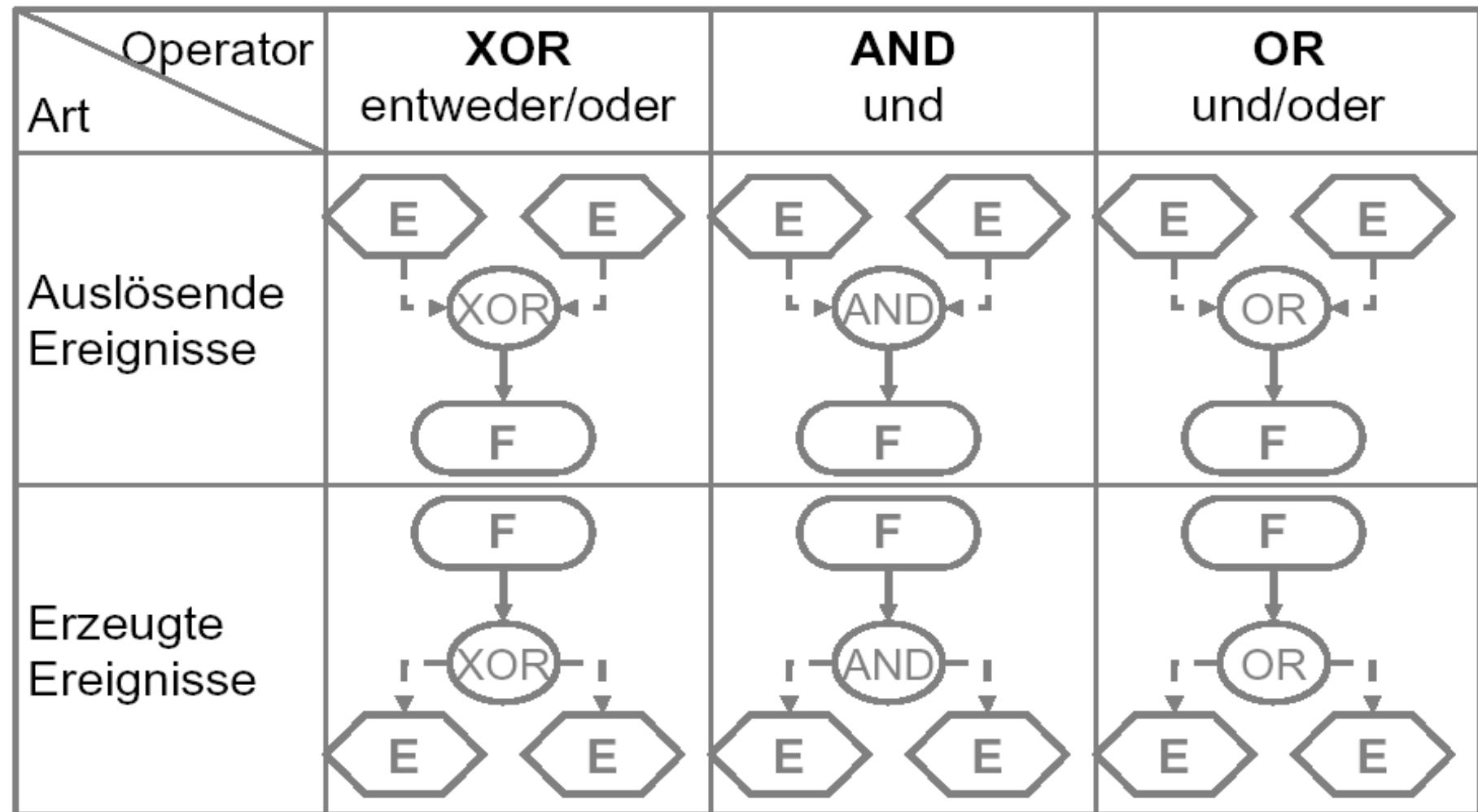
- Verknüpfungen können mehrere Eingänge oder Ausgänge haben (aber nicht beides!)



7 Verknüpfungsoperatoren bei EPK

■ Ereignisverknüpfung:

Mehrere Ereignisse werden mit einer Funktion verknüpft.

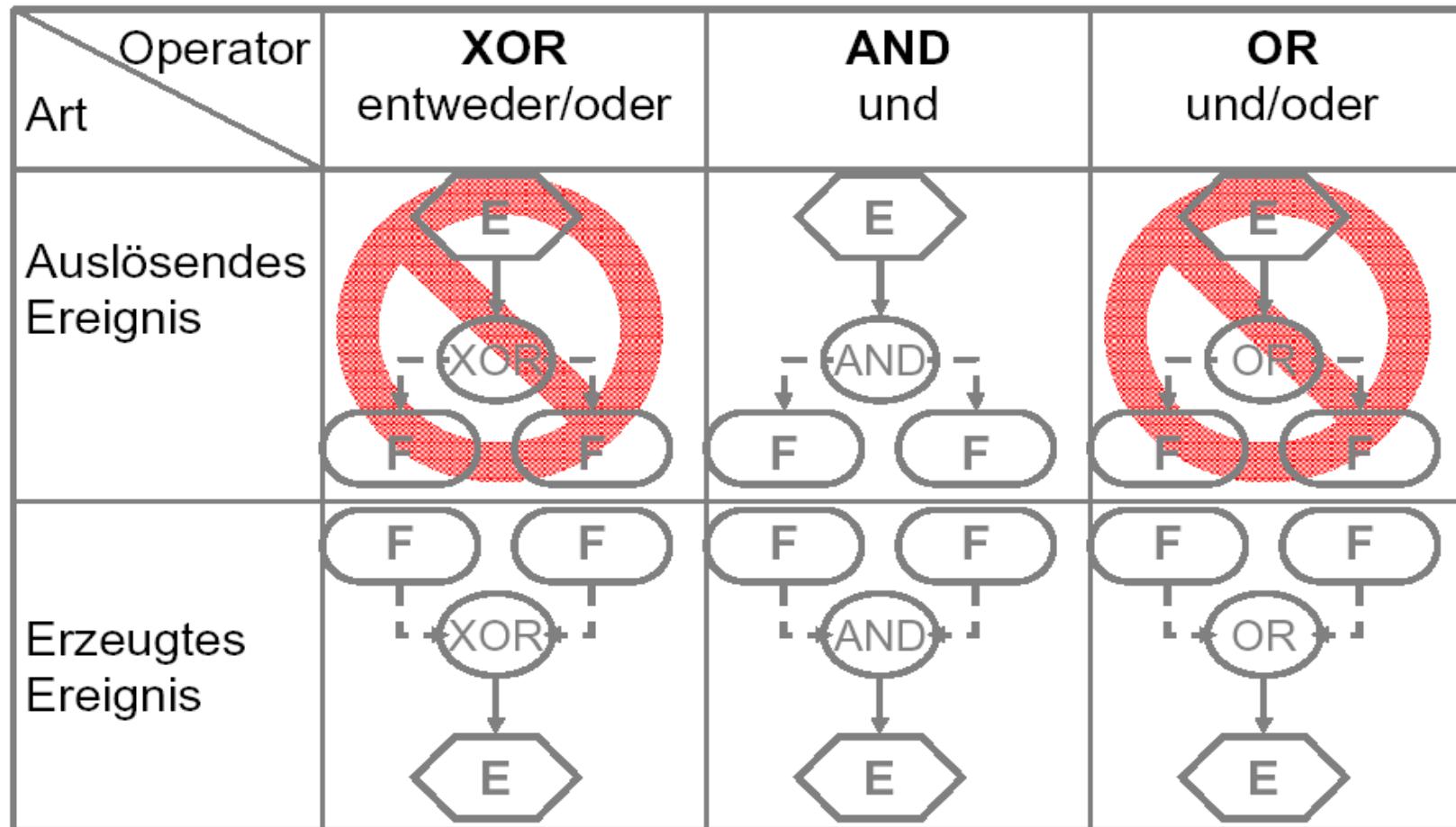


7 Verknüpfungsoperatoren bei EPK

■ Funktionsverknüpfung:

Mehrere Funktionen werden mit einem Ereignis verknüpft.

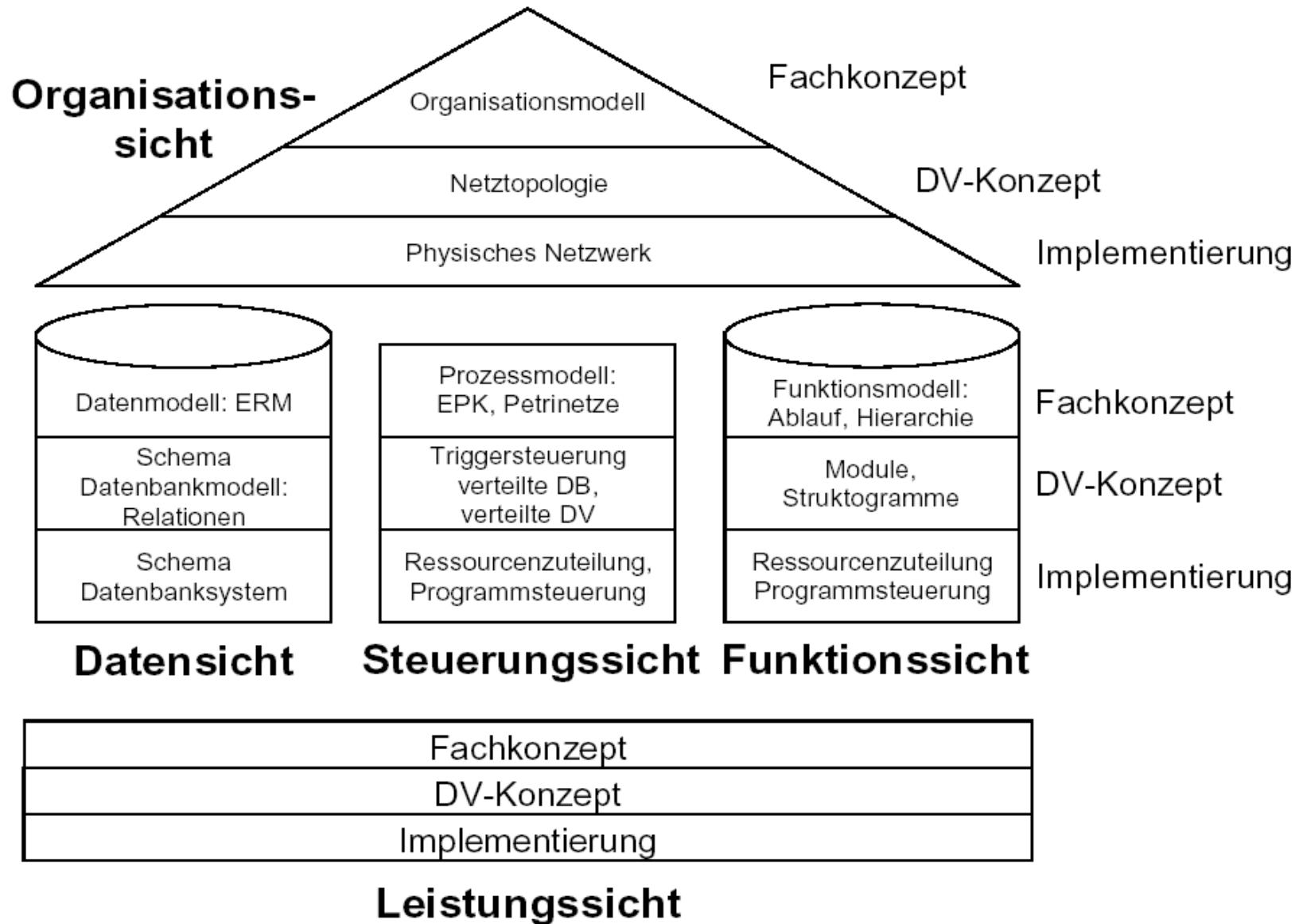
Verbot, XOR- bzw. OR-Konnektor anzuwenden, da Ereignisse als passive Elemente keine Entscheidungen treffen können.



8 Erweiterte EPK und ARIS

- **ARIS** = Architektur integrierter Informationssysteme,
- beschreibt die einzelnen Bausteine eines Informationssystems hinsichtlich
 - ihrer Art,
 - ihrer funktionalen Eigenschaften und
 - ihres Zusammenwirkens.
- Im Einzelnen werden mit ARIS ...
 - ... ein Rahmenkonzept (Architektur) zur vollständigen Beschreibung von Anwendungssoftware-Systemen angeboten,
 - ... in die Architektur die am geeignetsten erscheinenden Methoden zur Modellierung von Informationssystemen eingeordnet bzw. neue Methoden zur Geschäftsprozessbeschreibung entwickelt,
 - ... Tools zur Verwaltung von Anwendungswissen in Form von Referenzmodellen, zur Modellierung und Analyse von Anforderungen an Systeme sowie zur benutzerfreundlichen Navigation durch Modelle angeboten.

8 Erweiterte EPK und ARIS / ARIS-Haus



Fünf Sichten:

- **Funktionssicht**

- beinhaltet Funktionen, Aufgaben, Aktivitäten oder Vorgänge,
- Weiterhin Ziele der Funktionen sowie verwendete Anwendungssoftware,
- Struktur des Funktionsmodells gibt an, welche Funktionen in anderen Funktionen enthalten sind,

- **Organisationssicht**

- hier werden Ressourcenklassen in Form von Org.-Einheiten (z.B. Stelle, Abteilung, Bereich, ...) definiert,
- Auch maschinelle Ressourcen sowie Betriebsmittel, Software, etc.

- **Datensicht**

- beinhaltet Daten, die im Ablauf anfallen, sowie Ereignisse oder Nachrichten, die Vor- oder Nachbedingungen von Funktionen sind,

- **Leistungssicht**

- Modellierung von materiellen und immateriellen Input- und Output-Leistungen (Sach-, Dienst-, Informationsleistungen incl. Geldflüsse),

- **Steuerungs-/Prozesssicht**

- enthält die dynamischen Aspekte des Geschäftsprozess-Ablaufs, vollständige Prozessbeschreibung.

Drei Phasen:

Jede Sicht durchläuft drei Phasen, um ein Informationssystem zu erstellen.

- **Fachkonzept**

- formalisierte Beschreibung für spätere informationstechnische Realisierung, d.h. Funktionsmodelle, Datenmodelle, Leistungsmodelle sowie Prozessmodelle sind in den entsprechenden Sichten im Rahmen des Fachkonzepts hier einzuordnen,

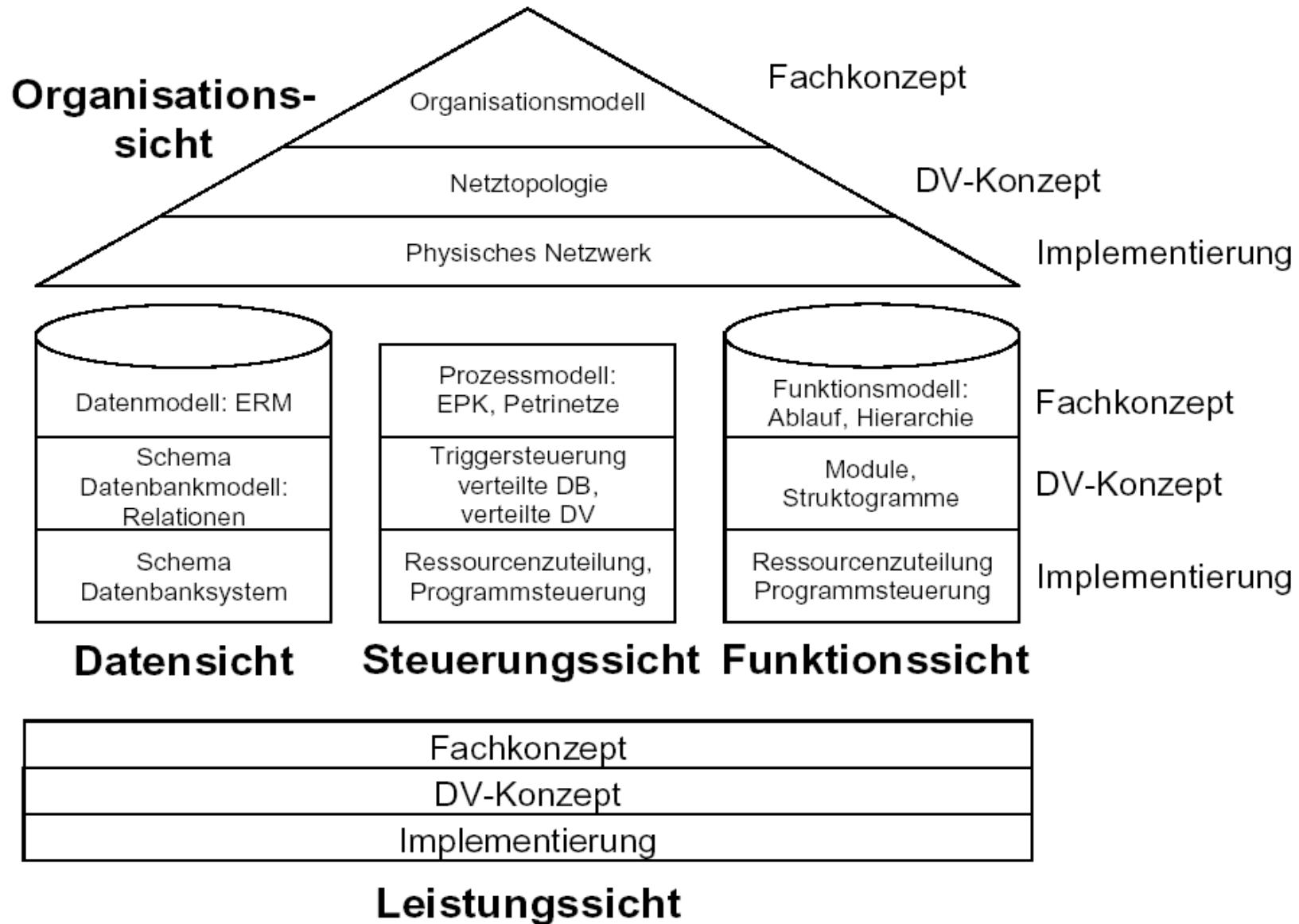
- **DV-Konzept**

- Schnittstelle zwischen Fachkonzept und technischer Implementierung,

- **Implementierung**

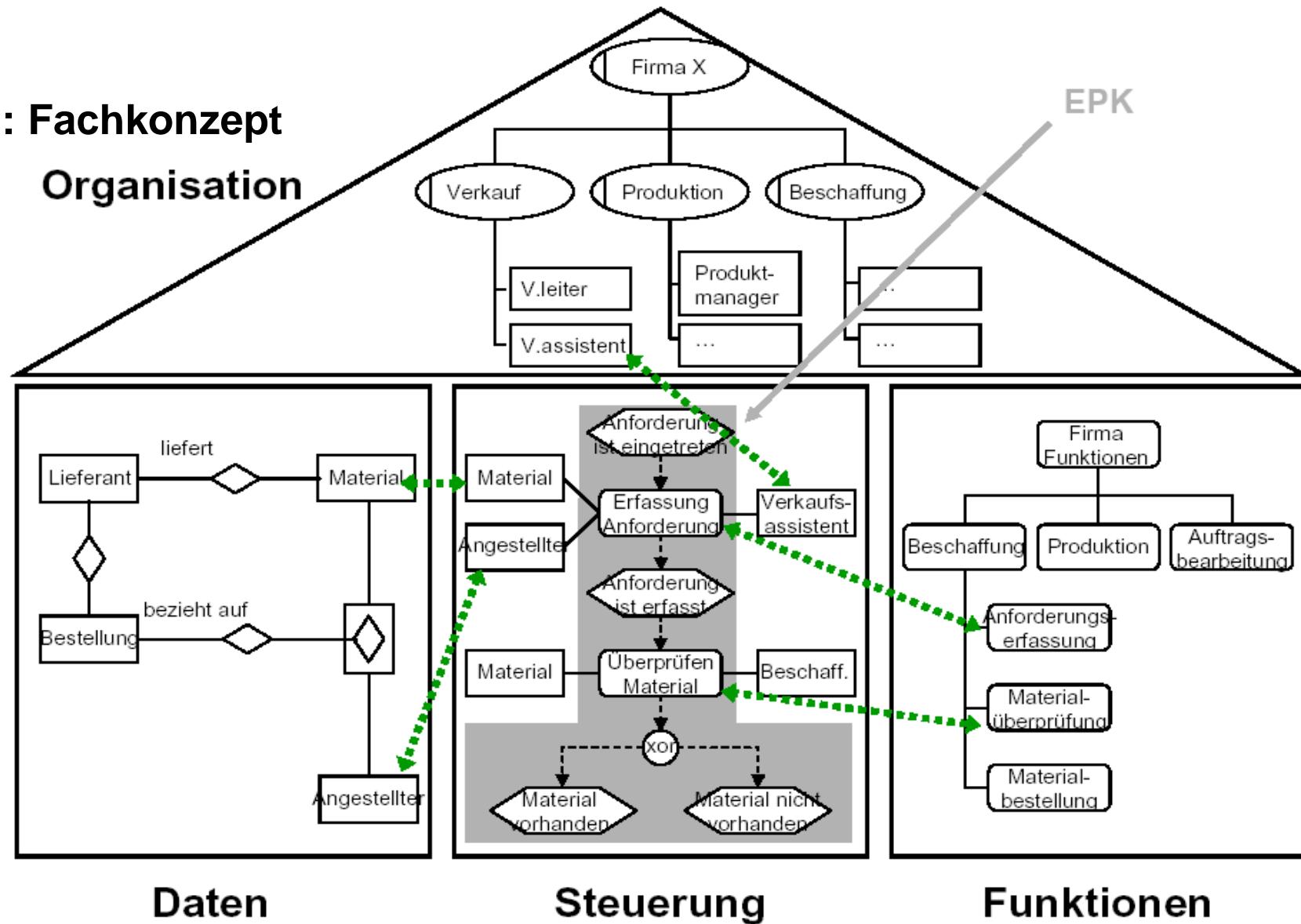
- Einsatz/Umsetzung konkreter Hard-Komponenten, Datenstrukturen, Softwarelösungen entsprechend den Anforderungen.

8 Erweiterte EPK und ARIS / ARIS-Haus



8 Erweiterte EPK und ARIS / ARIS-Haus

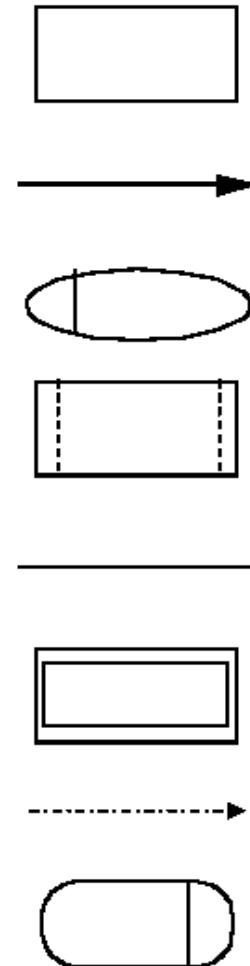
Beispiel: Fachkonzept Organisation



8 Erweiterte EPK und ARIS

Ergänzungselemente – Erweiterte EPK

- Datenobjekt
- Informationsfluss
- Organisatorische Einheit
- Anwendungssoftware
- Zuordnung von Ressourcen
- Leistungsobjekt
- Leistungsfluss
- Funktion wird durch EPK verfeinert



Den Funktionen können

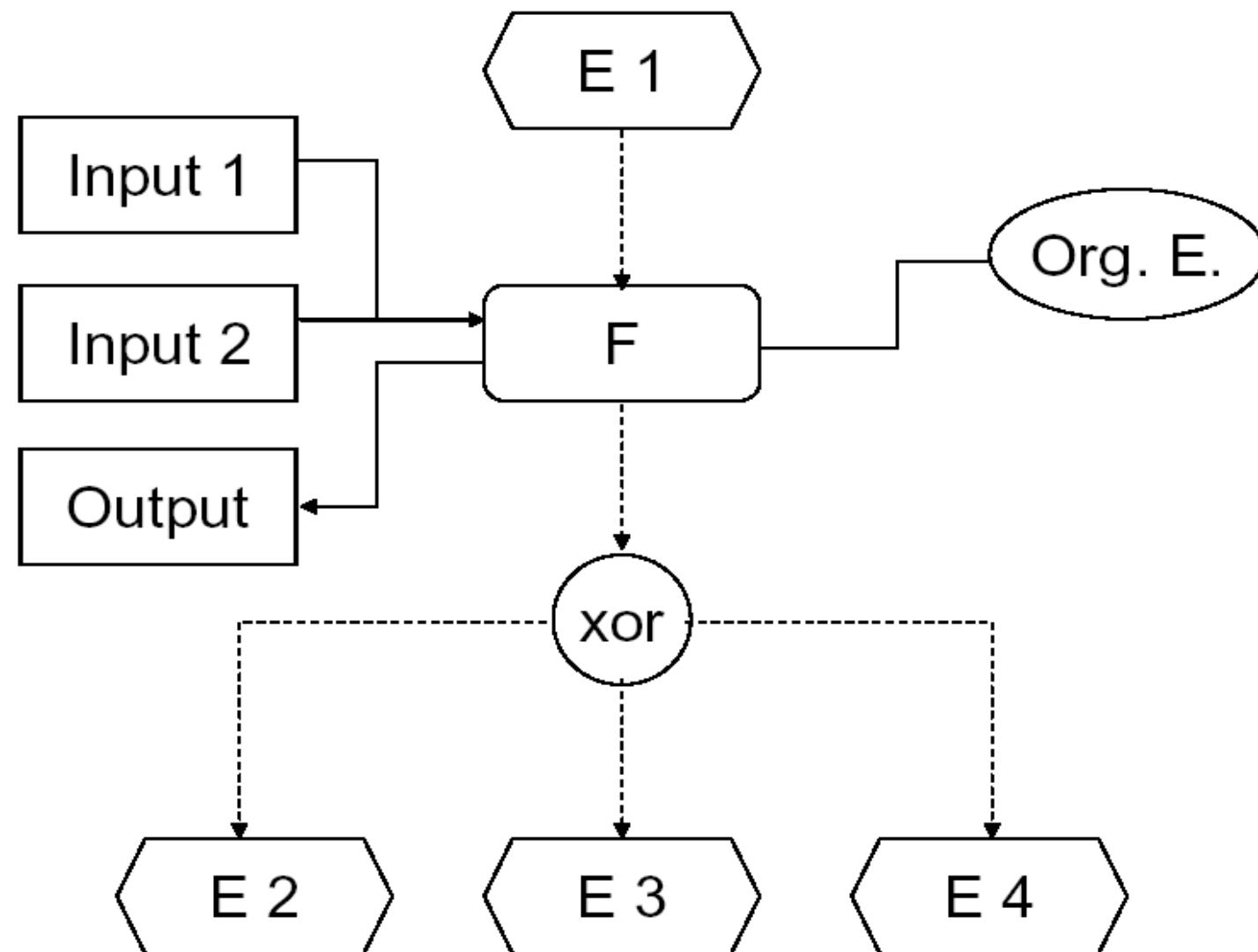
- die mit der Ausführung betrauten Organisationseinheiten sowie
- ein- und ausgehende Datenobjekte

zugeordnet werden.

Damit wird

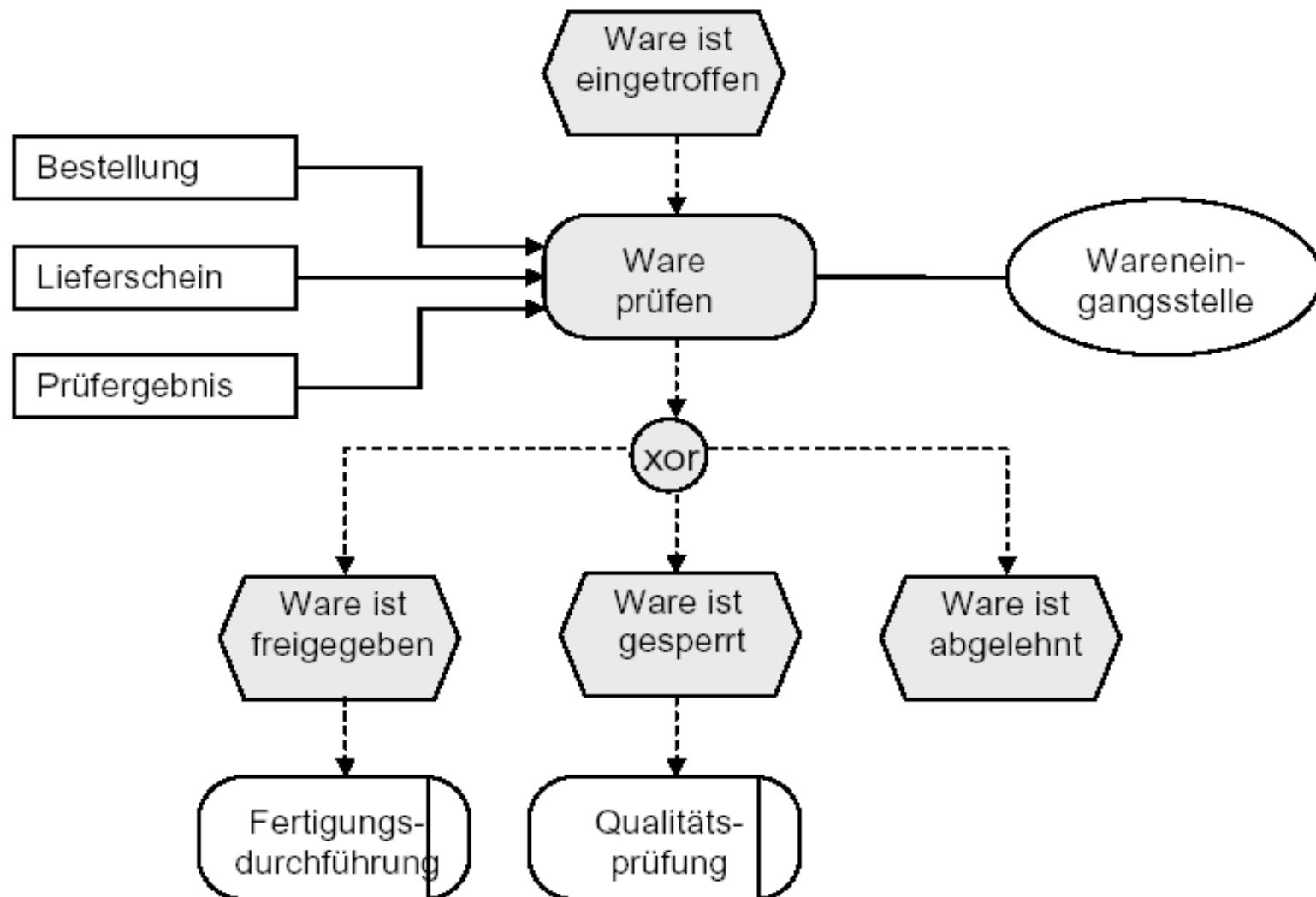
- die Prozessverfolgung über mehrere Organisationseinheiten hinweg möglich und
- zu jedem Informationsobjekt kann angegeben werden, von welcher organisatorischen Einheit es kommt, bzw. zu welcher organisatorischen Einheit es geschickt werden soll.

8 Erweiterte EPK und ARIS

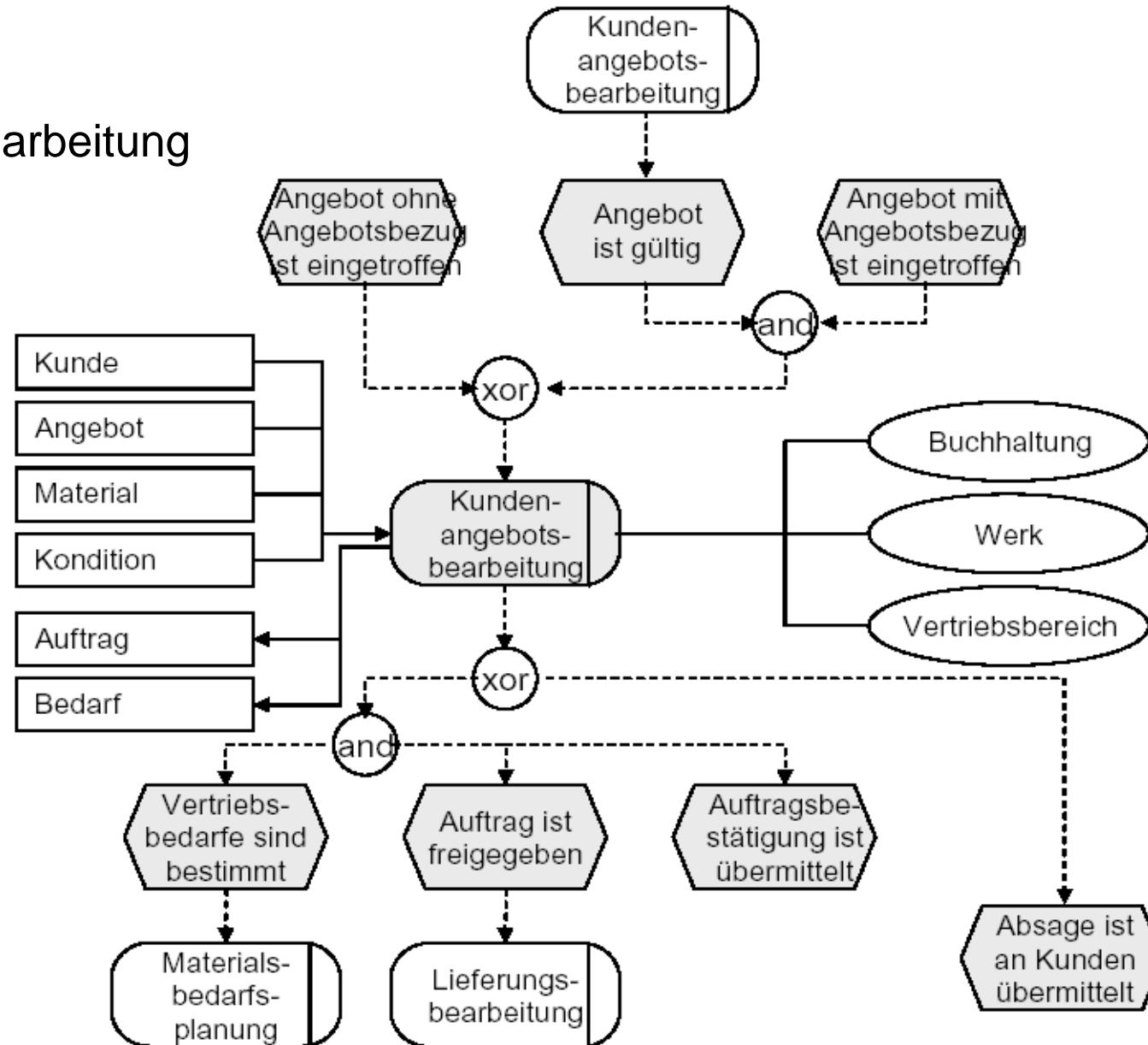


8 Erweiterte EPK und ARIS

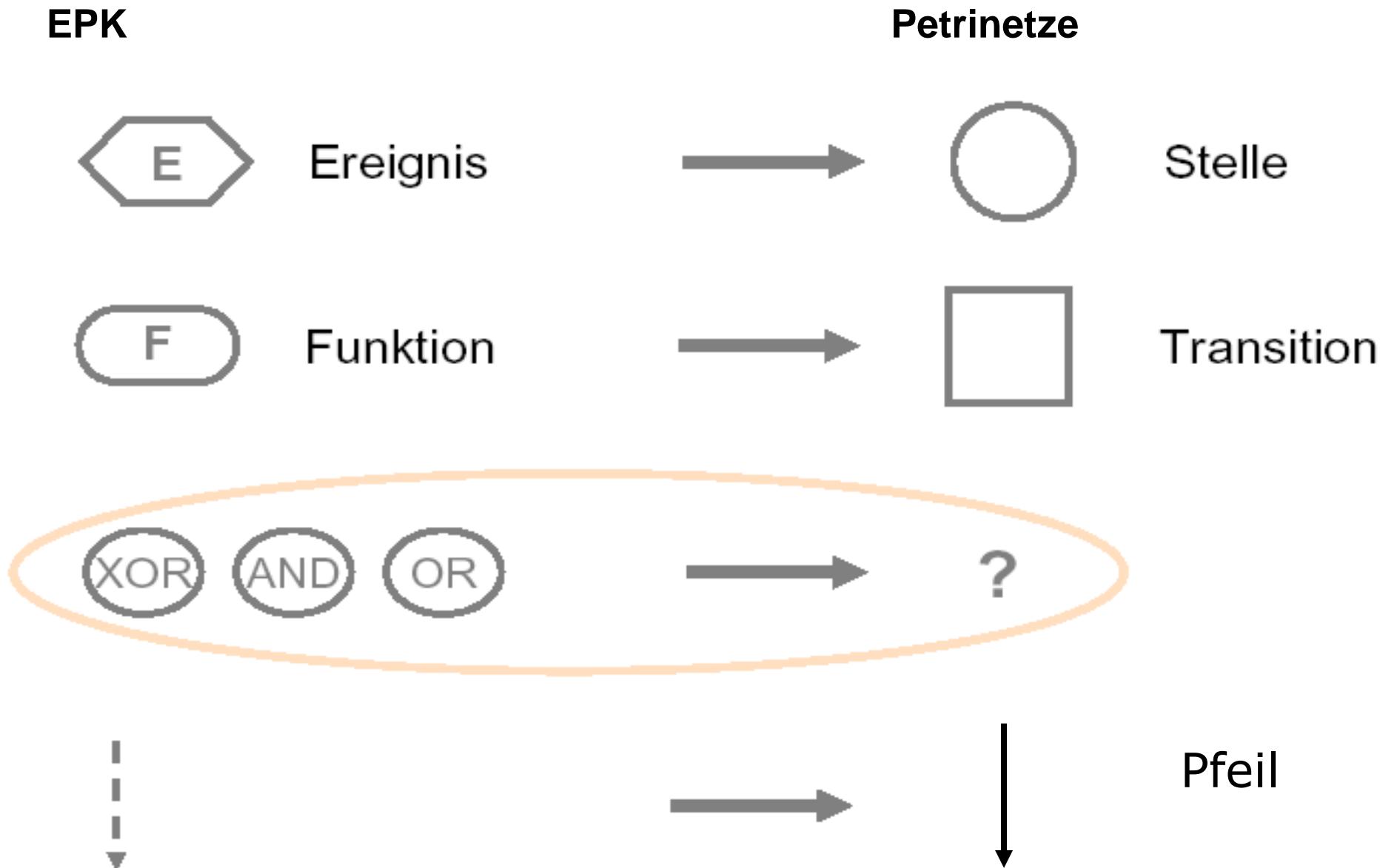
Beispiel 4: Wareneingangsbearbeitung



Beispiel 5: Auftragsbearbeitung

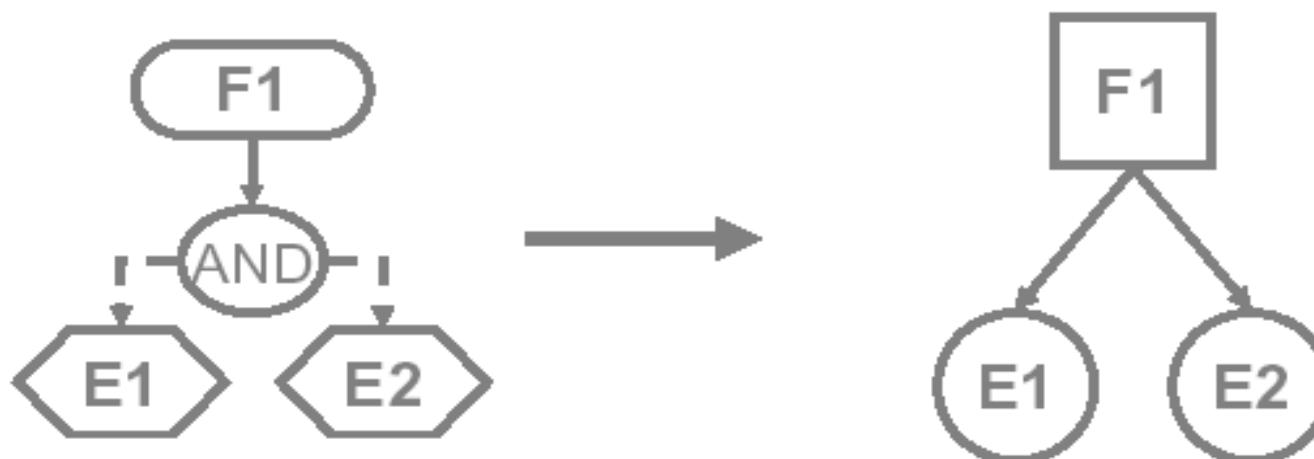
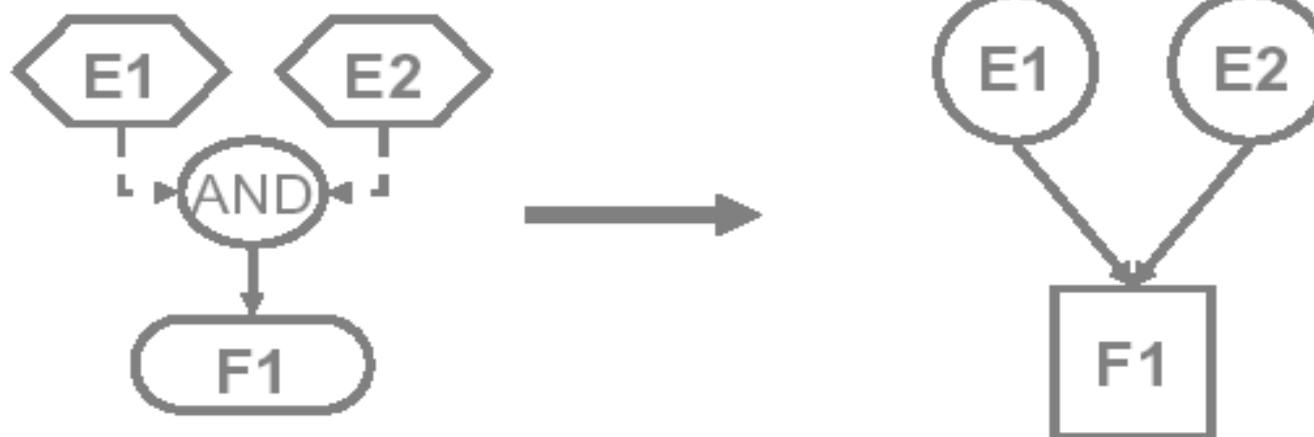


9 EPK vs. Petrinetze



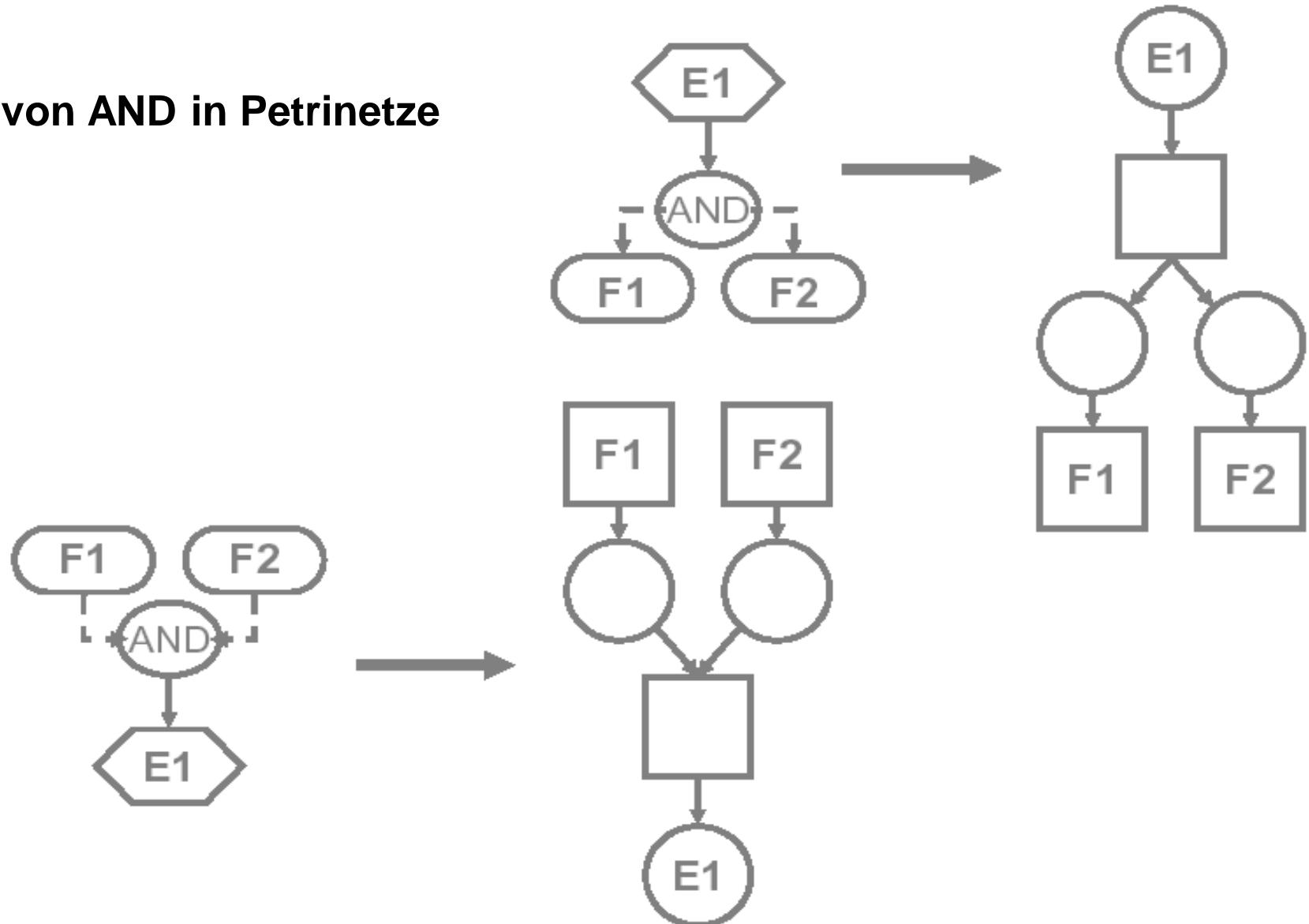
9 EPK vs. Petrinetze

Übersetzung von AND in Petrinetze



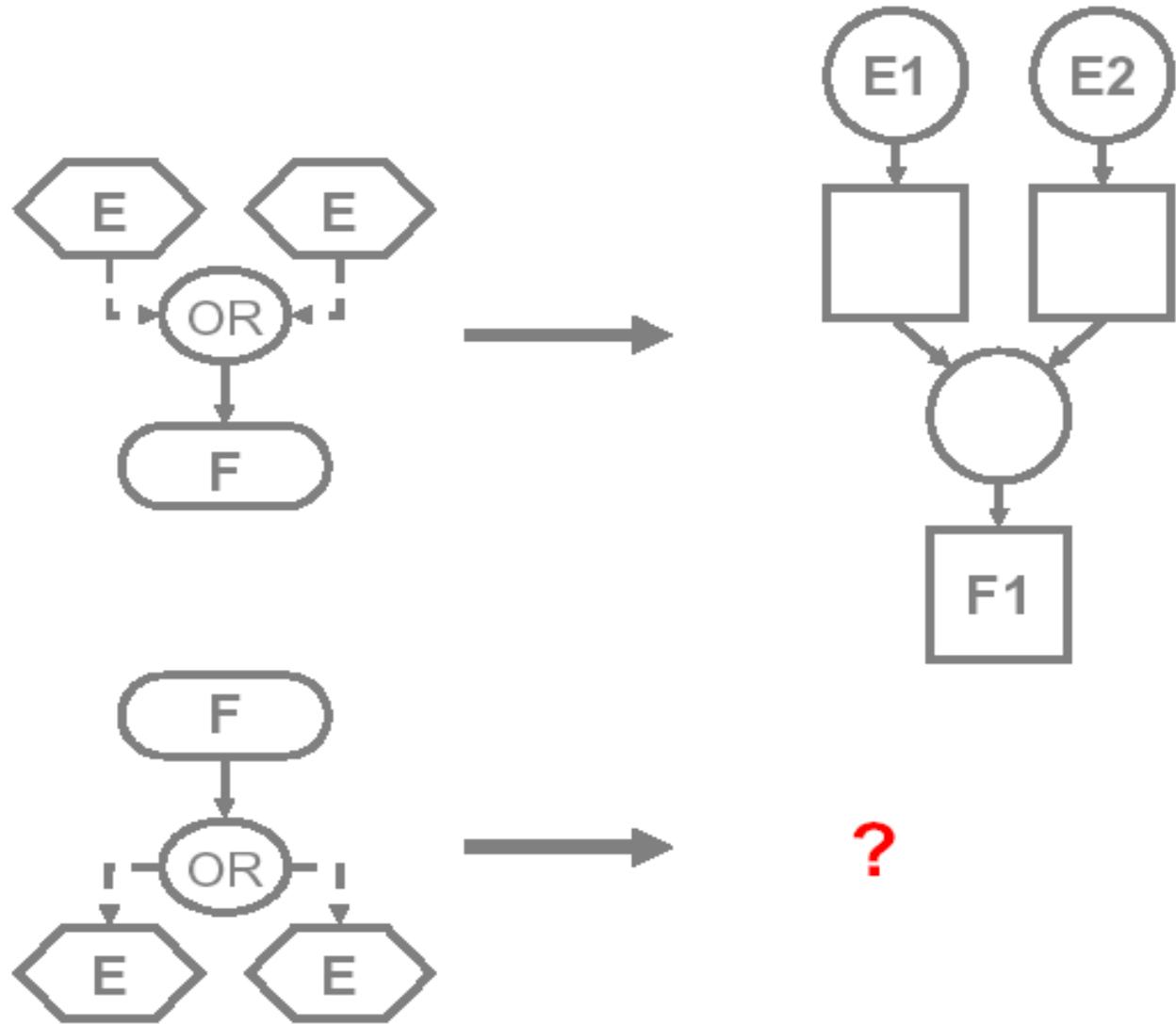
9 EPK vs. Petrinetze

Übersetzung von AND in Petrinetze



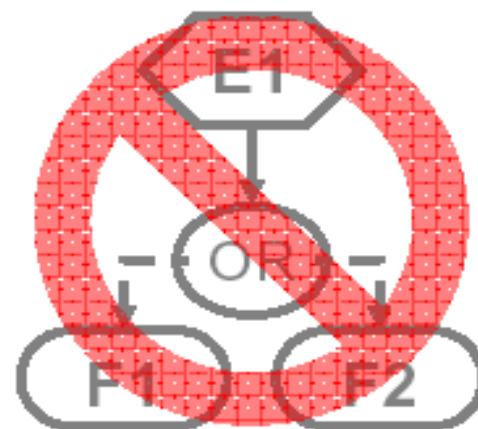
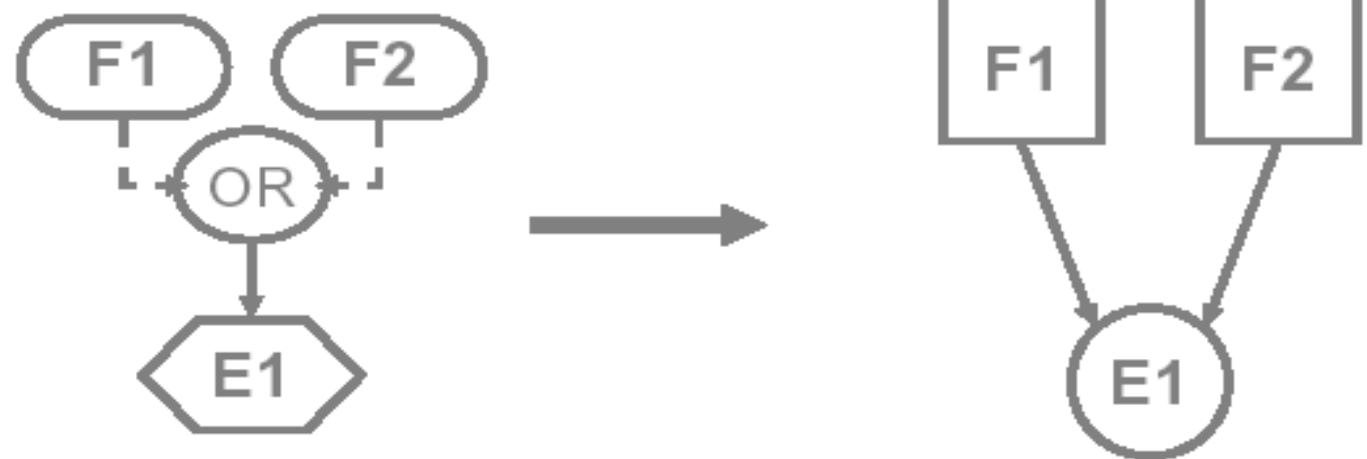
9 EPK vs. Petrinetze

Übersetzung von OR in Petrinetze



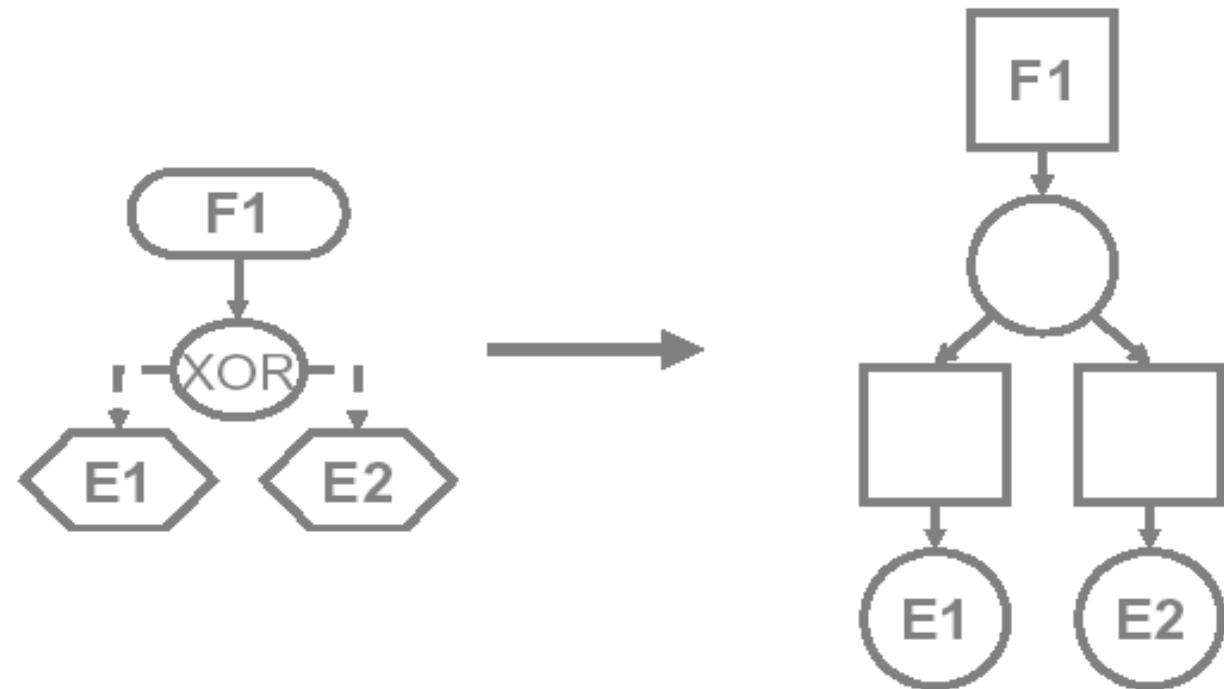
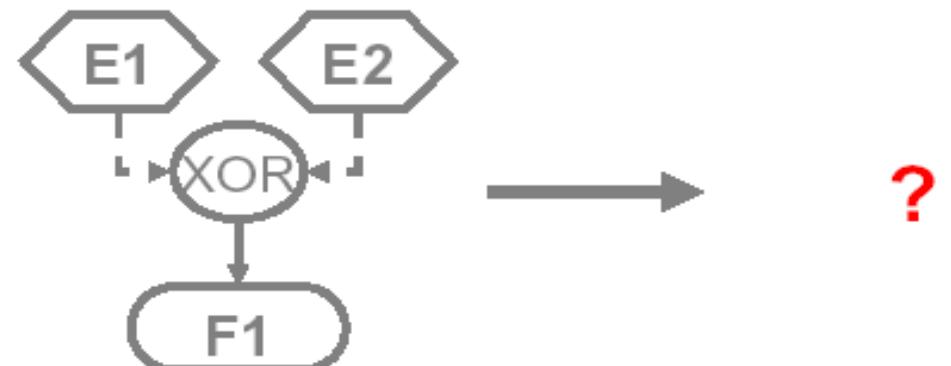
9 EPK vs. Petrinetze

Übersetzung von OR
in Petrinetze

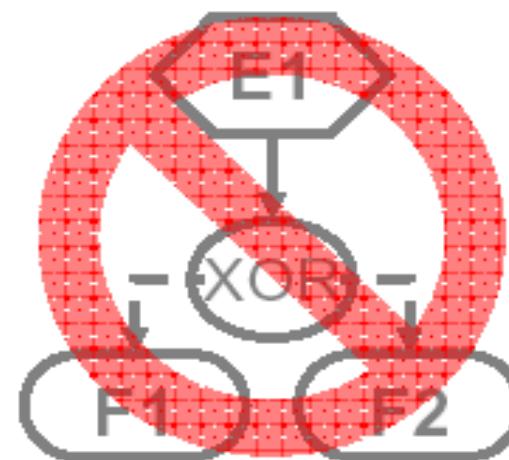
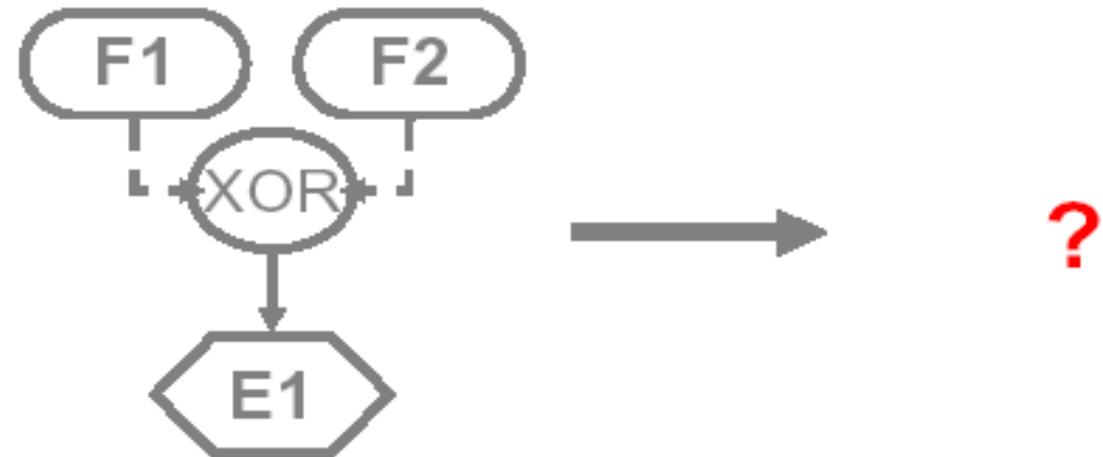


9 EPK vs. Petrinetze

Übersetzung von XOR in Petrinetze



Übersetzung von XOR in Petrinetze



Bewertung EPK im Hinblick auf Petrinetze

- hohe Anschaulichkeit und Sichtintegrationsfähigkeit (ARIS),
- Nur eine statische Sicht auf Prozess-Strukturen,
- Während sich Petrinetze sehr gut als Simulationssprache verwenden lassen, können die EPK diesen erhöhten methodischen Anforderungen ohne syntaktisch-semantische Erweiterungen nicht gerecht werden,
- Während Petrinetze eine umfangreiche und tiefe logisch-mathematische Fundierung besitzen, sind EPK syntaktisch und semantisch informell und besitzen noch keine hinreichenden Regeln für die Modellausführung.

Geschäftsprozess-Management

Prof. Dr.-Ing. Andreas Ittner

Email: ittner@hs-mittweida.de

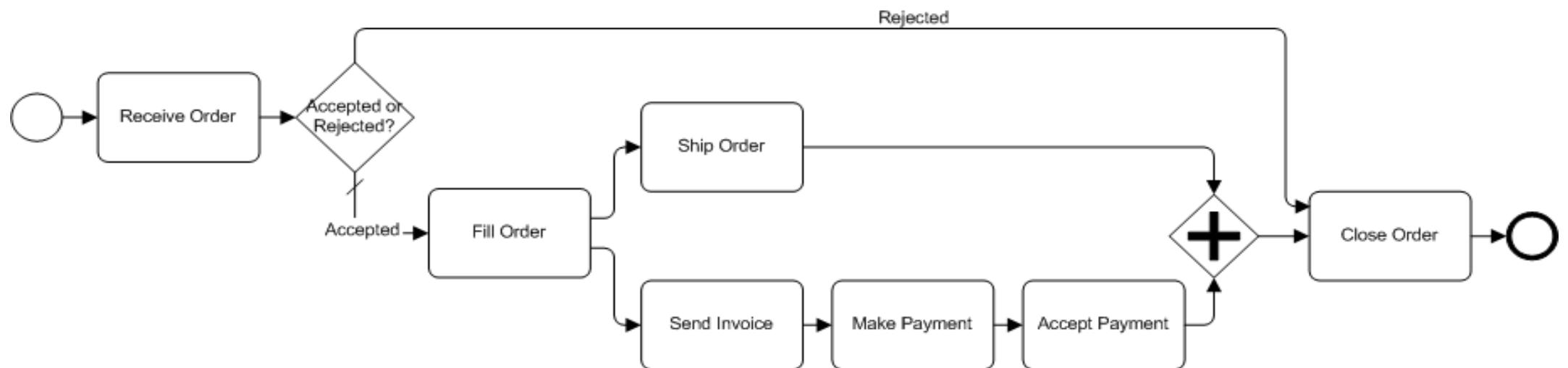
WWW: www.hs-mittweida.de/~ittner

Tel.: +49(0)3727-58-1288

Gliederung (vorläufig)

- Motivation
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs, **BPMN**, ...
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards

- Business Process Management (BPM)
- Business Process Management Initiative (BPMI)
- Business Process Model and Notation (BPMN)
- Business Process Execution Language (BPEL)



- Der BPMN-Standard
 - Einführung
 - Einordnung
 - Ziele
- Die Modellierungselemente
 - Kernelemente (Core Element Set)
 - Erweiterte Elemente (Extended Element Set)
- Business Process Diagramm Beispiele
- Ausführbare BPMN-Modelle (Mapping to WS-BPEL Sample)

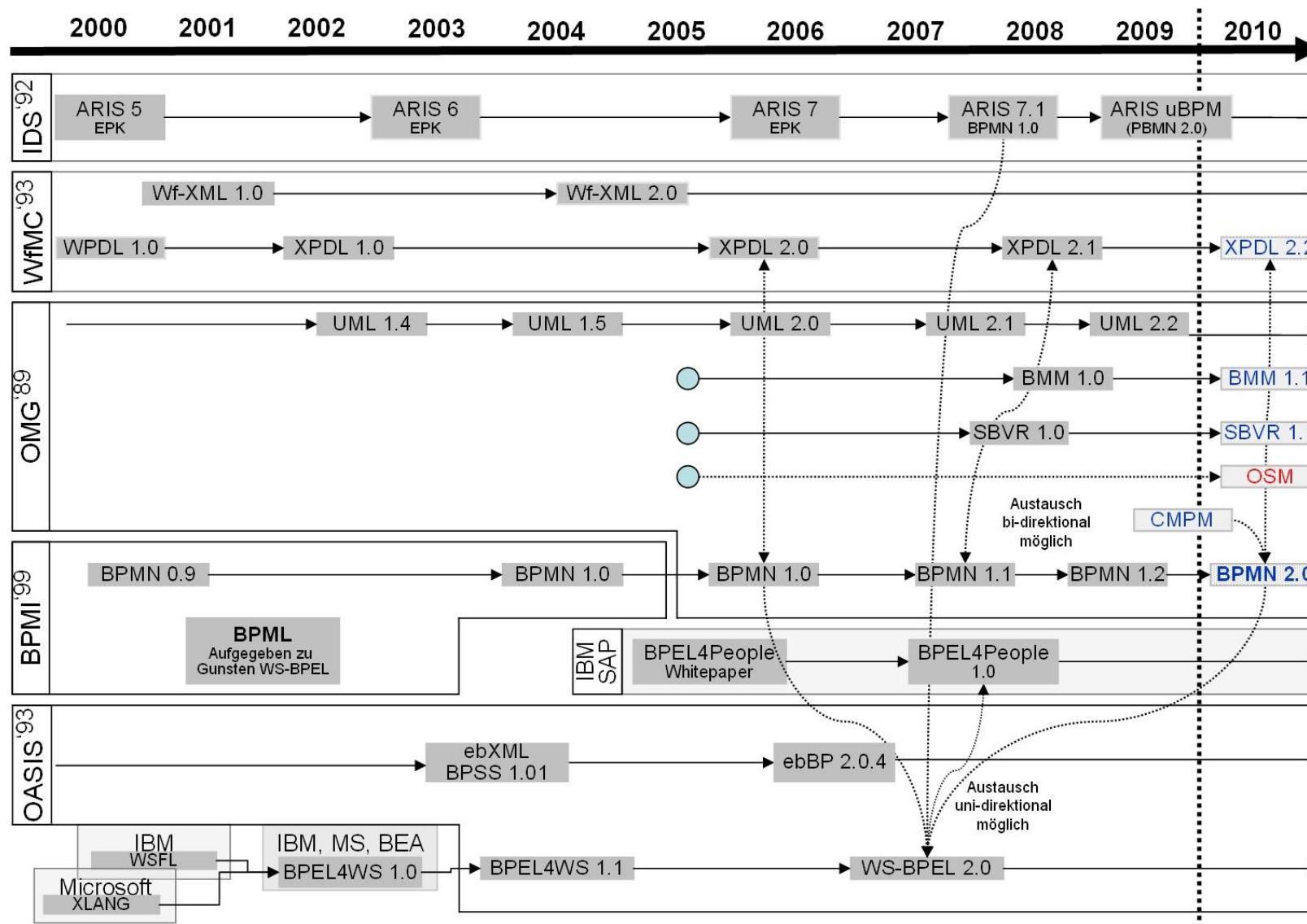
Hintergrund 2001 - 2005 (1)

- Gründung der Notation Working Group
 - Gründung im August, 2001, gegenwärtig ca.60 Mitglieder aus 35 Unternehmen, Organisationen und Einzelpersonen.
- BPMN 0.9 Draft
 - November, 2002, der BPMN 0.9 - Entwurf wurde der Öffentlichkeit vorgestellt,
- BPMN 1.0 Draft
 - August, 2003, der BPMN 1.0 - Entwurf wurde der Öffentlichkeit vorgestellt,
- BPMN 1.0
 - May, 2004, the BPMN 1.0 – Spezifikation wurde veröffentlicht.
 - Gegenwärtig gibt es ca. 30 Unternehmen, die die BPMN implementiert und ca. 5 Unternehmen, die momentan eine Implementierung entwickeln.
- Merger mit der OMG
 - Juni, 2005, BPMN 1.x wurde weiterentwickelt. BPMN 1.0 wurde in die OMG zur Standardisierung und kontinuierlichen Weiterentwicklung eingebracht.

Hintergrund 2005 - heute (2)

- BPMN 1.x Spezifikation (Maintenance Releases)
 - Produkt der BPMN FTF (Free Technology Foundation)
 - Fehler in der Spezifikation und Inkonsistenzen wurden behoben,
- BPMN 2.0
 - Anpassung an BPDM-Konzepte (Business Process Definition Metamodel), einschließlich Erweiterung bzgl. Choreografie
 - Start eines neuen RFP (Request for Proposal),
- BPMN Serialization
 - Entwicklung eines Metamodells (BPDM?) für BPMN, um Semantik bzgl. der Diagramm Information speichern und transportieren zu können,
 - Verwendung von XMI (XML Metadata Interchange), um Layout-Informationen zu speichern und zu transportieren,
- High-Level BPMN Erweiterungen.

Hintergrund 2000 - heute (3)



Dr. Martin Bartonitz, Nov. 2009

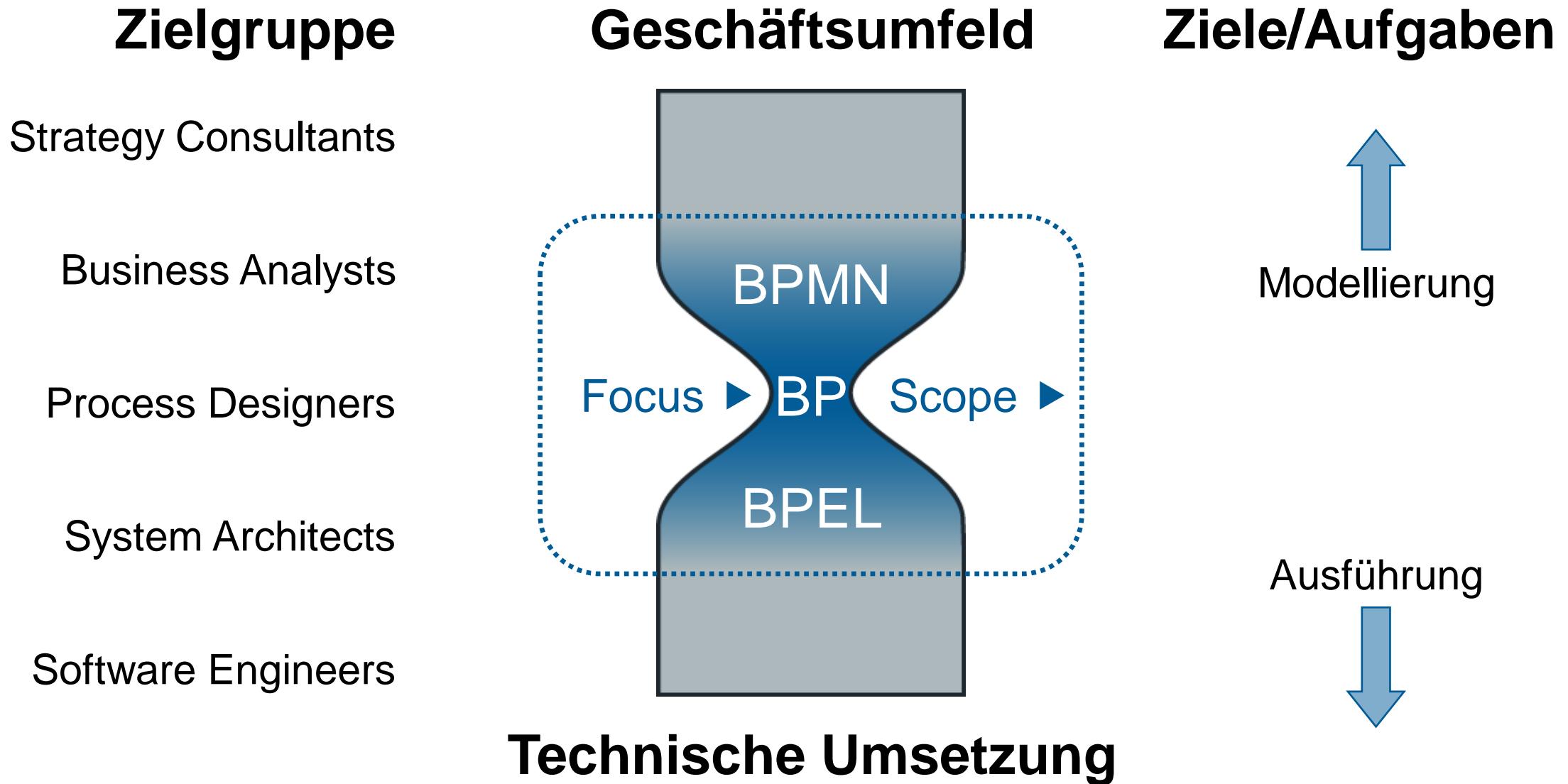
- Wer oder Was war BPMI (Business Process Management Initiative) ?
 - Es handelte sich dabei um eine non-profit Kooperation von Unternehmen unterschiedlicher Größe und aus unterschiedlichen Unternehmensbranchen (z.B. Adobe, Boeing, Borland, HP, IBM, SAP, Software AG, Sun, ...)
 - Ist 2005 in der OMG aufgegangen und somit keine eigenständige Organisation mehr.

Ziel (weiterhin): Technologien zu entwickeln, die es den Unternehmen ermöglicht, sich stärker prozessorientiert auszurichten.

- Erreicht werden soll dies durch die Etablierung von offenen Standards für Prozessdesign, Einsatz, Ausführung, Wartung und Optimierung.



BPM-„Sanduhr“: Geschäftsprozesse als Schnittpunkt zwischen Modellierung und Ausführung



- Was ist BPMN (Business Process Model and Notation)?
 - BPMN kann als ein formaler Mechanismus verstanden werden, der die Möglichkeit bietet, aus der Prozessnotation auf Business Level einen ausführbaren Geschäftprozess zu generieren.
 - Mit Hilfe der verschiedenen Diagrammelemente im BPMN können unterschiedlichste Typen von Geschäftsprozessen modelliert werden.

Ziel: Möglichst einfache und verständliche Darstellung und Modellierung von Geschäftsprozessen für den Nutzer zu erreichen.

- Grafische Modellierung von GPs mit einfacher und eindeutiger Lesbarkeit für alle Beteiligte:
 - Prozess-Modellierer, die auf fachlicher Ebene die GPs definieren,
 - Entwickler, die die modellierten GPs in einem IT-System implementieren
 - GP-Analysten, die die implementierten Prozesse steuern, kontrollieren, bewerten und optimieren,
- Visualisierung XML-basierter Sprachen für die Ausführung von GPs (z.B. BPEL4WS - Business Process Execution Language for Web Services) durch eine einfache grafische Notation
- BPMN schließt die „Lücke“ zwischen der Prozess-Modellierung auf der einen Seite und der Implementierung auf der anderen.

Die beiden Ziele führen zu einem Konflikt

- **Ziel 1:** Einfache und eindeutige Lesbarkeit
 - tendenziell eher niedriger Detaillierungsgrad in einem BPMN-Diagramm angestrebt, um Lesbarkeit zu bewahren,
- **Ziel 2:** Visualisierung aller für die automatisierte Ausführung eines GPs notwendigen Details
 - hoher Detaillierungsgrad nötig, um alle Anforderungen abbilden zu können
- **Lösung:** Aufteilen der Modellierungselemente in
 - Kernelemente (Core Element Set) und
 - Erweiterte Elemente (Extended Element Set).

Weitere Gliederung

- Business Process Diagram Elements
 - Core Set of Diagram Elements
 - Complete Set of Diagram Elements
- Business Process Diagram Samples
 - Normal Flow
 - B2B Modeling
 - Exception Handling
 - Compensation Handling
 - A Complex Process
- Mapping to WS-BPEL Sample (ehemals BPEL4WS)

Weitere Gliederung

- Business Process Diagram Elements
 - Core Set of Diagram Elements
 - Complete Set of Diagram Elements
- Business Process Diagram Samples
 - Normal Flow
 - B2B Modeling
 - Exception Handling
 - Compensation Handling
 - A Complex Process
- Mapping to WS-BPEL Sample (ehemals BPEL4WS)

Weitere Gliederung

- Business Process Diagram Elements
 - **Core Set of Diagram Elements**
 - Complete Set of Diagram Elements
- Business Process Diagram Samples
 - Normal Flow
 - B2B Modeling
 - Exception Handling
 - Compensation Handling
 - A Complex Process
- Mapping to WS-BPEL Sample (ehemals BPEL4WS)

- Die Menge der Kernelemente lässt sich in 4 Kategorien unterteilen:
- **Ablaufobjekte** (Flow Objects) modellieren das Verhalten eines GPs
- **Verbindungsobjekte** (Connecting Objects) verbinden die Ablaufobjekte miteinander oder mit anderen Informationen
- **Schwimmbahnen** (Swim Lanes) gruppieren fachlich zusammengehörige Modellierungselemente grafisch und
- **Artefakte** (Artifacts).

Core Set of Diagram Elements - Ablauf- und Verbindungsobjekte

Events



Sequence Flow



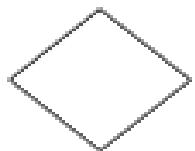
Activities



Message Flow



Gateways



Association



- Die zentralen Modellierungs-elemente (core set) erlauben die einfache Entwicklung simpler Business Process Diagramme.
- Sie sind den meisten Business Analysten vertraut und z.B. Flowchart-Diagrammen sehr ähnlich.

Core Set of Diagram Elements, Events and Activities

Events

Unter einem Event wird verstanden, dass etwas passiert während des Ablaufs des Prozesses.

Start



Unterbrechnung
(Zwischenereignis)



Stop



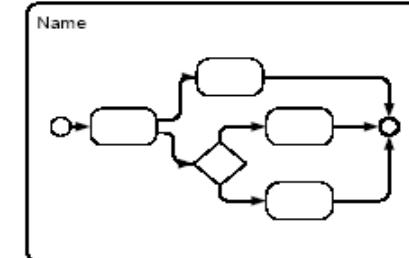
Activities

Eine Aktivität ist eine Aufgabe, die im Rahmen des Geschäftsprozesses ausgeführt wird.

Task

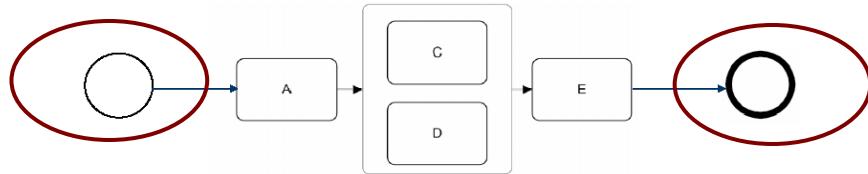


Prozess /
Subprozess



Core Set of Diagram Elements, Events

- Ein **Ereignis** ist etwas, das während des Prozess-Ablaufs passiert/eintritt.
- Diese Ereignisse beeinflussen den Prozess-Fluss, sie haben normalerweise einen Trigger oder ein Ergebnis.
- Sie können den Fluss starten, unterbrechen oder beenden.



- Start-, End- und Zwischenereignisse,
- Verschiedene Auslösertypen, die einen Prozess auslösen, im Ablauf beeinflussen oder beenden,
- Verwendung von Start-, und Endereignis ist abhängig vom Prozesslevel,
- Vorbedingung: End-Event braucht Start-Event

Core Set of Diagram Elements, Activities

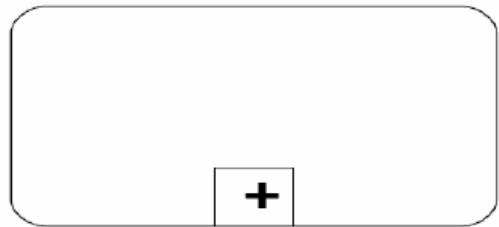
- Eine **Aktivität** ist eine Arbeitsaufgabe, die während eines Geschäftsprozesses (GP) ausgeführt wird.
- Eine Aktivität kann atomar oder zusammengesetzt (compound) sein.



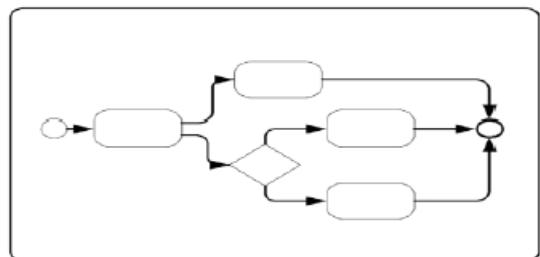
- Die Aktivitätstypen, die Teil des Prozess-Modells sind, können
 - Processes,
 - Sub-Processes, and
 - Tasks sein.

Core Set of Diagram Elements, Activities

- Ein Unterprozess (sub-process) kann auch in ausgeklappter (expanded) Form dargestellt werden, um Details von untergeordneten Aktivitäten zu zeigen.



collapsed



expanded

Core Set of Diagram Elements, Connections und Gateways

Connections

Es gibt drei Arten von Connections:

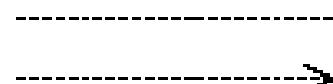
Die **Sequence Flow** zeigt wie die Aktivitäten innerhalb des Prozesses in ihrer Reihenfolge abgearbeitet werden.



Die **Message Flow** zeigt, wie Nachrichten zwischen zwei Teilnehmern (repräsentiert durch Pools die senden bzw. empfangen) fließen.



Die **Association** wird dazu benutzt, um zusätzliche Informationen oder Artifacts mit dem Prozess-Fluss zu verbinden.



Gateways

Gateways sind Modellierungselemente, mit denen Flüsse (Verzweigungen und Zusammenführungen) gesteuert werden.

Gateways

Exclusive Decision/Merge (XOR)

Data-Based



Event-Based



Inclusive Decision/Merge (OR)



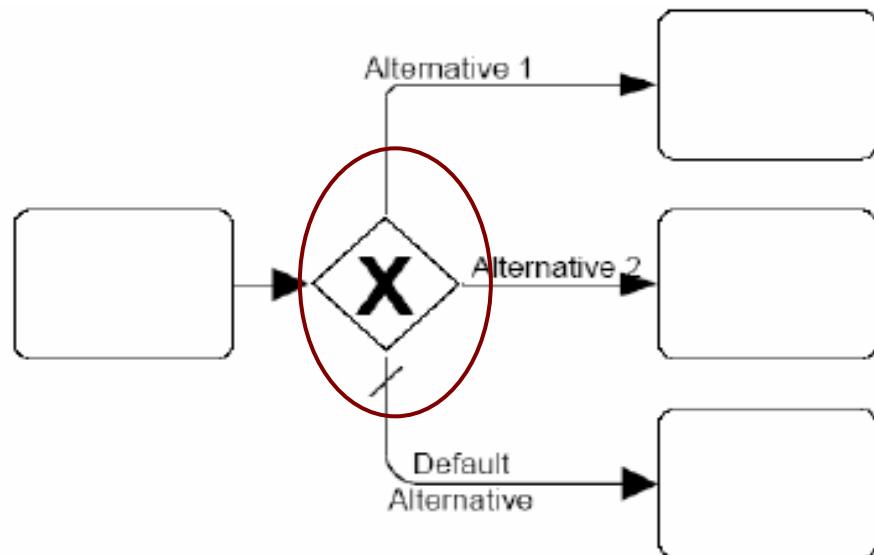
Complex Decision/Merge



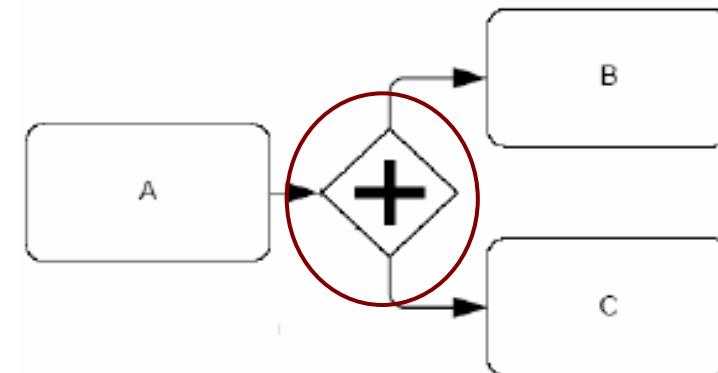
Parallel Fork/Join (AND)



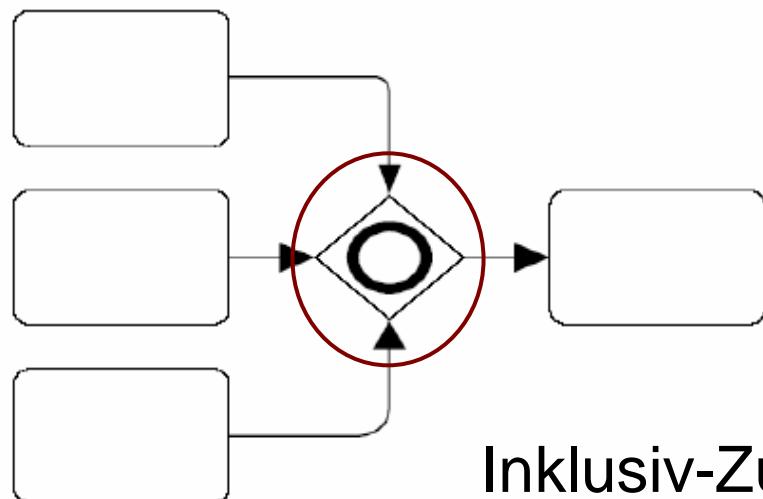
Complete Set of Diagram Elements, Gateways (data-based)



Exklusiv-Entscheidung

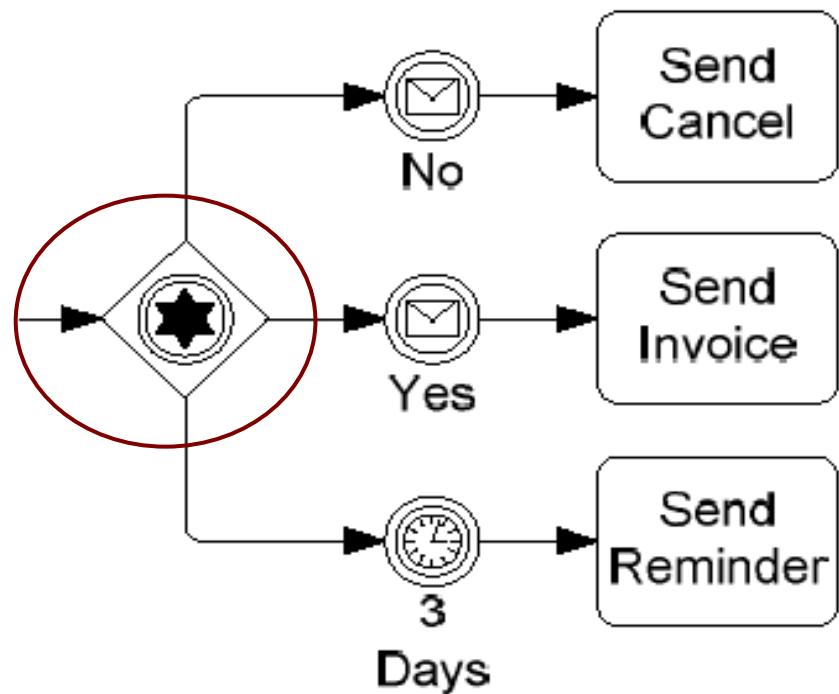


Parallele Verzweigungen



Inklusiv-Zusammenführungen

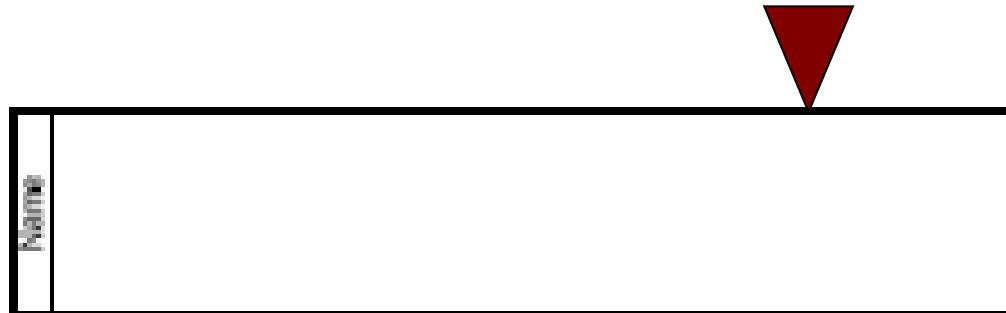
Complete Set of Diagram Elements, Gateways (event-based)



- Exclusive-Gateway, event-based,
- Verzweigung basiert nicht auf Bedingungen (data-based), sondern auf Ereignissen, die in diesem Prozess-Schritt auftreten können,
 - das zuerst auftretende Ereignis „gewinnt“.

Pools / Lanes

Pools repräsentieren Teilnehmer in einem Prozess, z.B. Unternehmen.

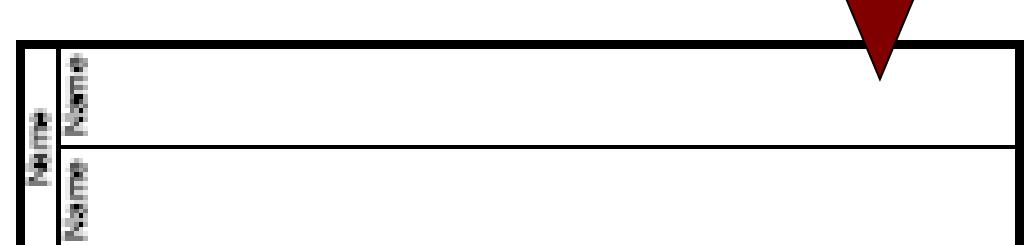


Pools sind Container, die eine Menge von Aktivitäten enthalten. Verbindungen zu anderen Teilnehmern (Pools) können nur über Nachrichtenfluss zustande kommen.

Pools können zur Abbildung von Geschäftsprozessen zwischen unterschiedlichen Unternehmen genutzt werden.

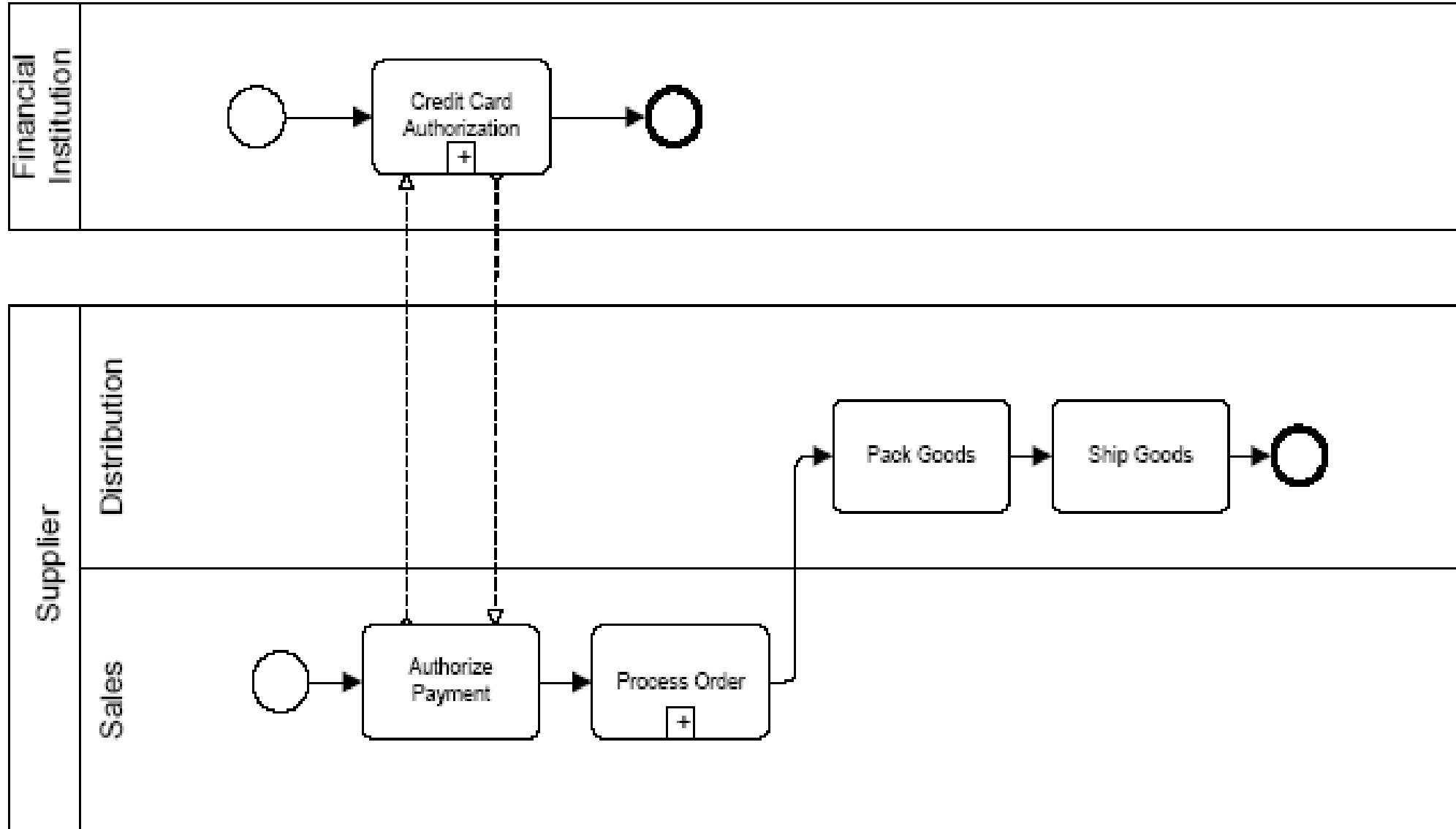
Core Elemente vom BPMN (3)

Pools können entlang ihrer Ausdehnung wiederum in Lanes unterteilt werden.



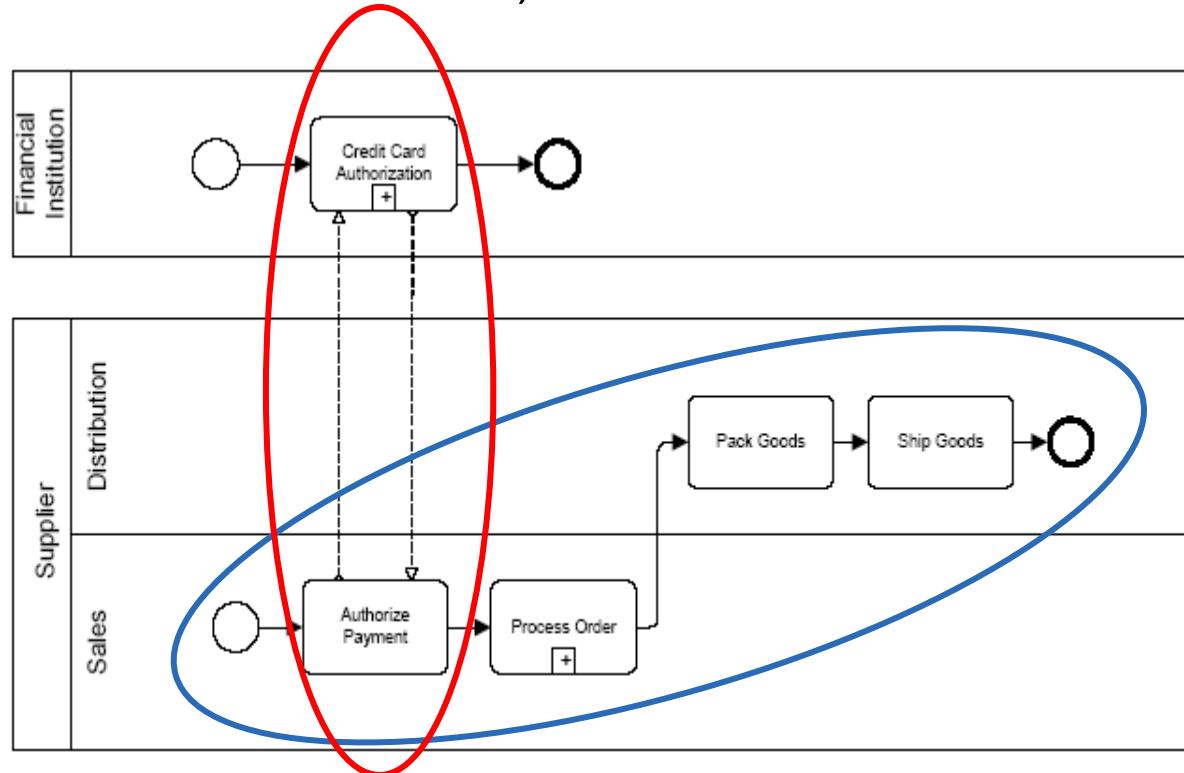
Lanes (Bahnen) repräsentieren Org.Einheiten, z.B. Abteilungen eines Unternehmens oder bilden Rollen (z.B. Manager, Abteilungsleiter) ab.

Beispiel für Schwimmbahnen (Swim Lanes)



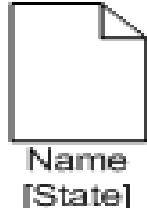
Orchestration vs. Choreography

- **Orchestration:** Workflow, interner Prozess, privater Prozess, Prozesse, die intern in einer spezifischen Organisation ablaufen (in einem einzigen Pool)
- **Choreography:** Kollaboration, globale Prozesse, B2B-Processes (beschreibt die Interaktion zwischen ≥ 2 Business Entities (die jeweils als Pools modelliert werden, d.h. Message Flow between the Pools).



Artifacts

Data Object



Name
[State]

Text Annotation

Text Annotation Allows a Modeler to provide additional Information

Group



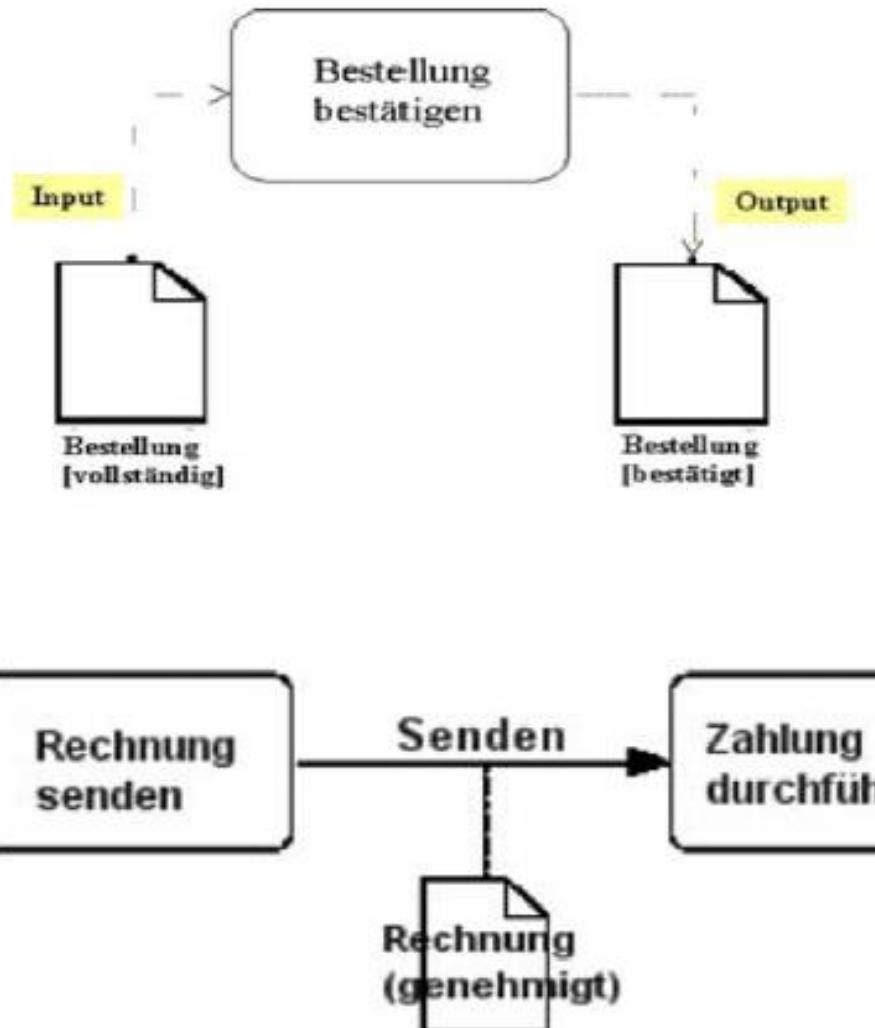
- **Data Objects** sind Fluss-Objekte, die lediglich Information darüber, wie Dokumente, Daten oder andere Objekte verwendet und aktualisiert werden während des Prozess-Ablaufs.

■

- **Text Annotations** sind ein Instrument für den Modellierer, um zusätzliche Informationen für den Leser/Nutzer des BPMN Diagramms zur Verfügung zu stellen.

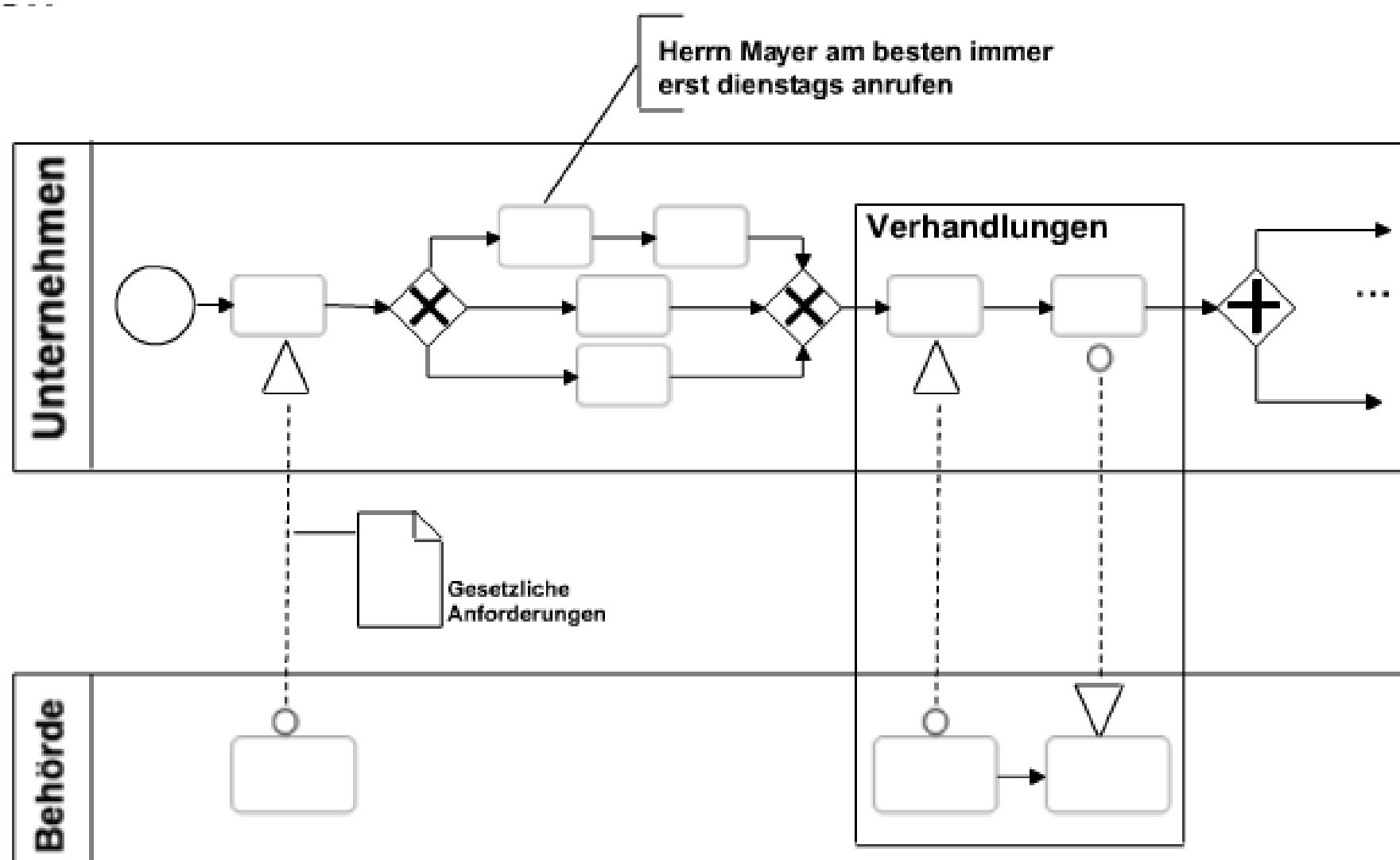
- **Groups** dienen dazu, um Aktivitäten auch grafisch-gruppiert zu organisieren/ darzustellen.

Beispiel für Artefakte (1/2)



- Beispiel: Dokument
 - Dokument mit einer gefalteten rechten oberen Ecke,
 - Das Modellieren solcher Elemente ist optional,
 - beeinflussen nicht den Prozessablauf,
 - geben aber detaillierter Auskunft über die Aktivität
- welche Bedingungen benötigt werden damit diese ausgeführt werden kann und was sie produziert

Beispiel für Artefakte (2/2)

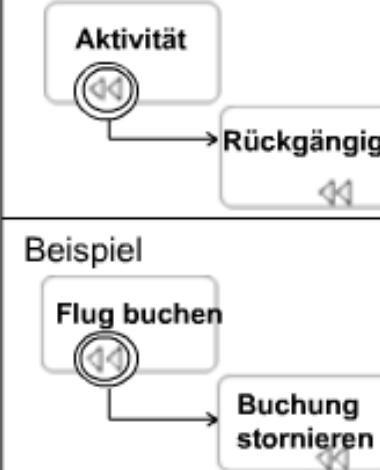


Weitere Gliederung

- Business Process Diagram Elements
 - Core Set of Diagram Elements
 - **Complete Set of Diagram Elements**
- Business Process Diagram Samples
 - Normal Flow
 - B2B Modeling
 - Exception Handling
 - Compensation Handling
 - A Complex Process
- Mapping to WS-BPEL Sample (ehemals BPEL4WS)

Complete Set of Diagram Elements (erweiterte Elemente)

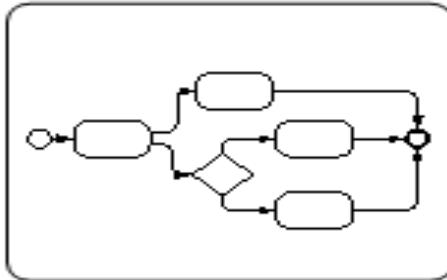
■ Erweiterung der Ablaufobjekte: Aktivitäten

Name	Beschreibung	Notation
Loop-Aktivität (Activity Looping)	Mehrfache, sequentielle Ausführung der Aktivität. Die Instanzen der Aktivität laufen hintereinander ab.	
Mehrfache Instanzen	Eine Aktivität kann mehrfach instantiiert werden. Die Instanzen laufen parallel ab.	
Kompensation (Compensation)	Kompensationen machen ausgeführte Aktivitäten rückgängig. Auslöser ist immer ein <i>Compensation Intermediate</i> -Ereignis. Eine Kompensation stellt somit einen rückwärtigen Prozessfluss dar, der zu einem bereits verlassenen Zustand zurückkehrt.	

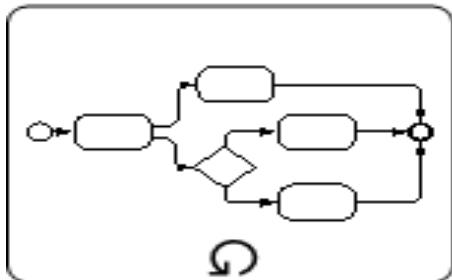
Complete Set of Diagram Elements (erweiterte Elemente)



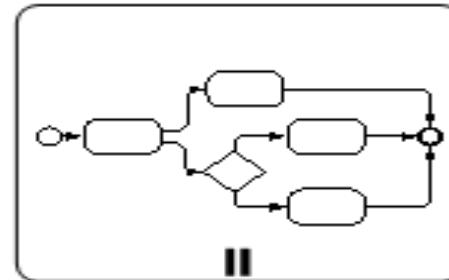
**Sub-Process
(Expanded)**



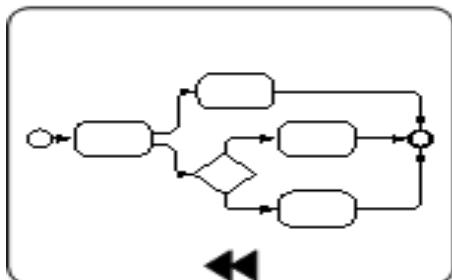
Loop



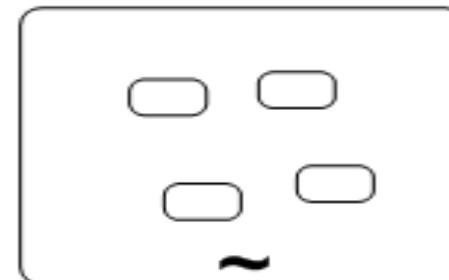
Multiple Instance



Compensation



Ad-Hoc



Modellierungsvorschriften für Aktivitäten

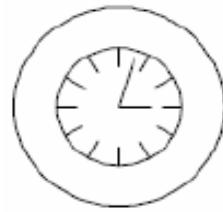
- Eine Aktivität kann mehrere eingehende Sequenzverbinder haben.
- Einer solcher Aktivität sollte ein Entscheidungsoperator vorgeschaltet werden, um den Ablauf kontrollieren zu können.
- Mehrere Sequenzverbinder können aus einer Aktivität herausführen.

Complete Set of Diagram Elements (erweiterte Elemente)

- Erweiterung der Ablaufobjekte: **Ereignisse**
- Ereignisse können um Typen erweitert werden,
- Verhältnisse, unter denen ein Ereignis auftritt, können so detaillierter beschrieben werden,
- Ausdrucksstärke, aber auch die Komplexität der GP-Diagramme kann steigen!,
- Nicht jeder Ereignistyp eignet sich als Start-, Zwischen- oder End-Ereignis

	Start	Intermediate	End
Message			
Timer			
Error			
Cancel			
Compensation			
Conditional			
Link			
Terminate			
Multiple			

Complete Set of Diagram Elements (erweiterte Elemente)



- Auslösertypen für Start-Ereignisse
 - Nachricht, Timer, Regel, ...,



- Auslösertypen für Zwischen-Ereignisse
 - Nachricht, Timer, Regel, ...,



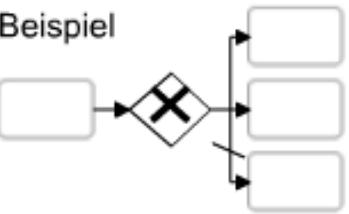
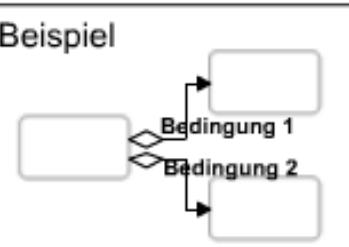
- Auslösertypen für End-Ereignisse
 - Nachricht, Ausnahme, Beenden...

Modellierungsvorschriften für Ereignisse

- Nur Ereignisse abbilden, die einen Einfluss auf den Prozessablauf haben,
- Prozesse sollten mindestens ein Start- und ein End-Ereignis haben,
- Start-Ereignis darf keinen eingehenden Sequenzverbinder haben,
- von jedem Start-Ereignis geht mindestens ein Sequenzverbinder aus,
- End-Ereignis muss Ziel mindestens eines Sequenzverbinder sein; es darf aber nicht die Quelle eines solchen sein,
- Zwischenereignisse haben genau einen ausgehenden Sequenzverbinder.

Complete Set of Diagram Elements (erweiterte Elemente)

■ Erweiterung der Ablaufobjekte: **Verbindungsobjekte**

Name	Beschreibung	Notation
Standardfluss (Default Flow)	Modellierung eines Standardflusses, der genau dann ausgeführt wird, wenn keine der anderen Bedingungen von ODER- oder XOR-Entscheidungen erfüllt sind.	 Beispiel 
Bedingter Sequenzverbinder (Conditional Sequence Flow)	Steuerung des Ablaufes über Bedingungen an den Sequenzverbindern. Ist die Bedingung erfüllt, wird die verbundene Aktivität ausgeführt. Semantisch gleichwertig zu einem ODER-Entscheidungsoperator.	 Beispiel 
Fehlerfluss (Exception Flow)	Ablauf im Fehlerfall, also wenn ein <i>Intermediate Error-Ereignis</i> vorliegt.	

Sequenzverbinder: Welche Verbindungen sind möglich?

Nach	○	+/-	□	◇	○	○
Von	○	ja	ja	ja	ja	ja
	+/-	ja	ja	ja	ja	ja
	□	ja	ja	ja	ja	ja
	○	ja	ja	ja	ja	ja
	○					

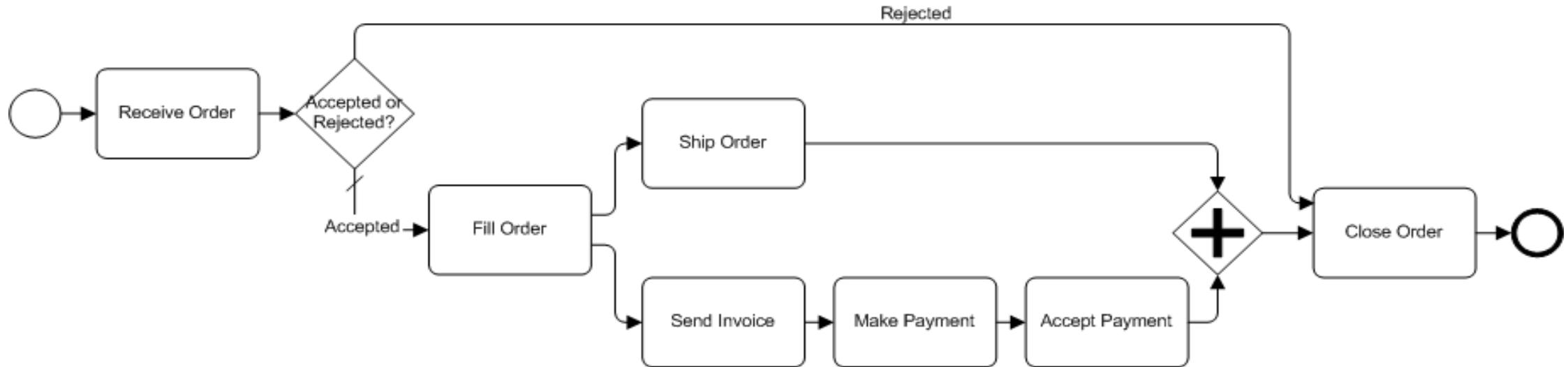
Nachrichtenfluss: Welche Verbindungen sind möglich?

Nach Von						
	ja	ja	ja	ja	ja	
	ja	ja	ja	ja	ja	
	ja	ja	ja	ja	ja	
	ja	ja	ja	ja	ja	
	ja	ja	ja	ja	ja	

Gliederung

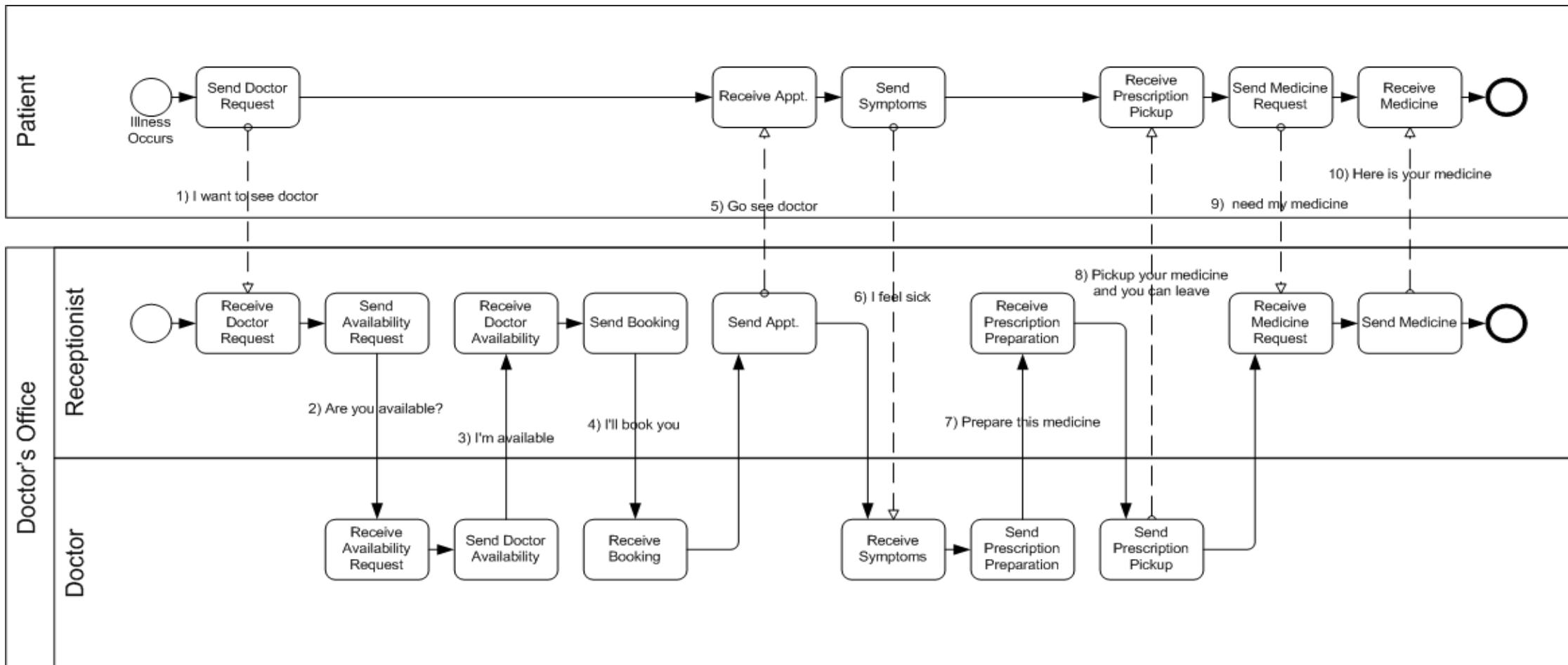
- Business Process Diagram Elements
 - Core Set of Diagram Elements
 - Complete Set of Diagram Elements
- Business Process Diagram Samples
 - Normal Flow
 - B2B Modeling
 - Exception Handling
 - Compensation Handling
 - A Complex Process
- Mapping to WS-BPEL Sample (ehemals BPEL4WS)

Normal Flow



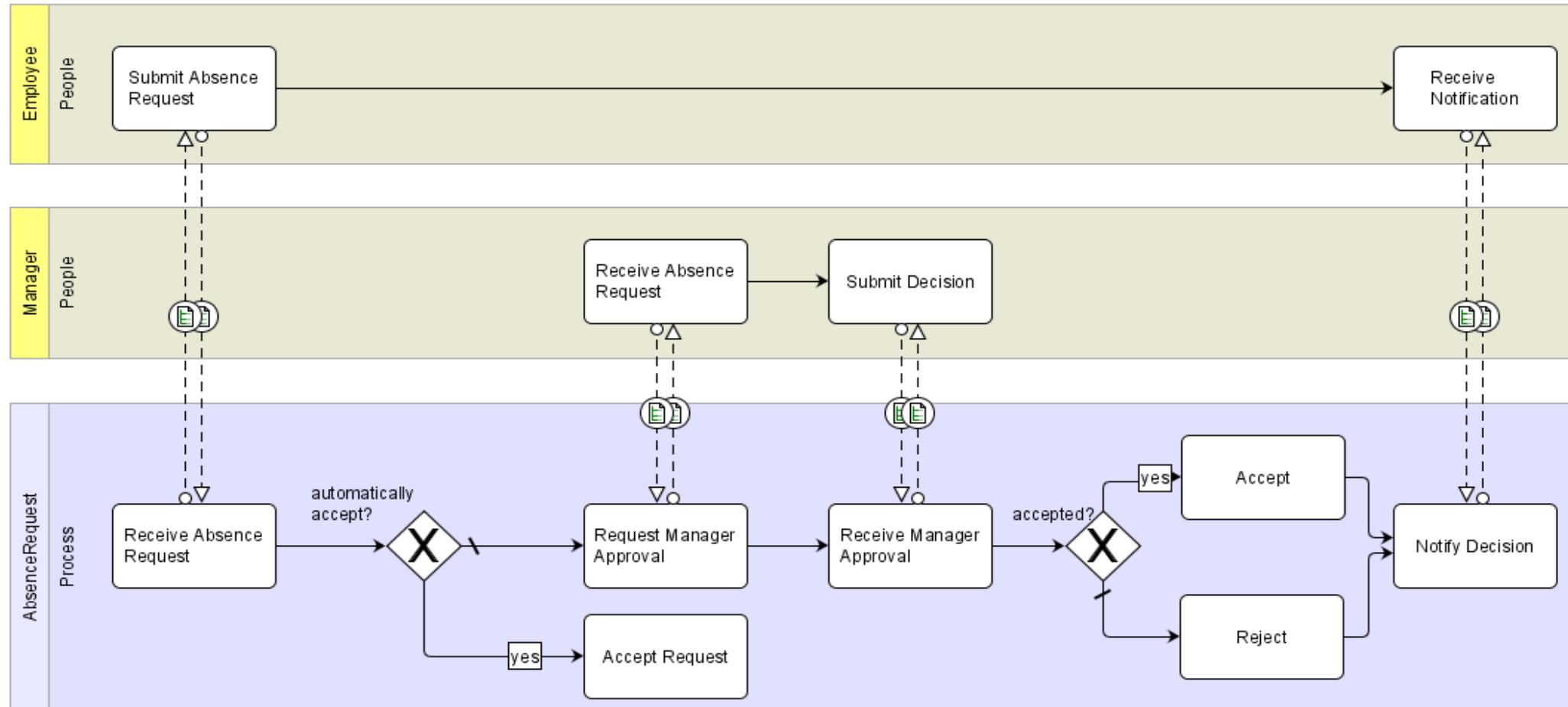
- Ein normaler Sequenz-Fluss beginnt mit einem Start Ereignis und verläuft über verschiedene Aktivitäten (auch Alternativen und parallele Pfade) bis zu seinem End Ereignis.
- Ein „normal flow“ enthält keine Ausnahme- (exception flow) oder Ersatzbehandlungen (compensation flow).

B2B Modeling



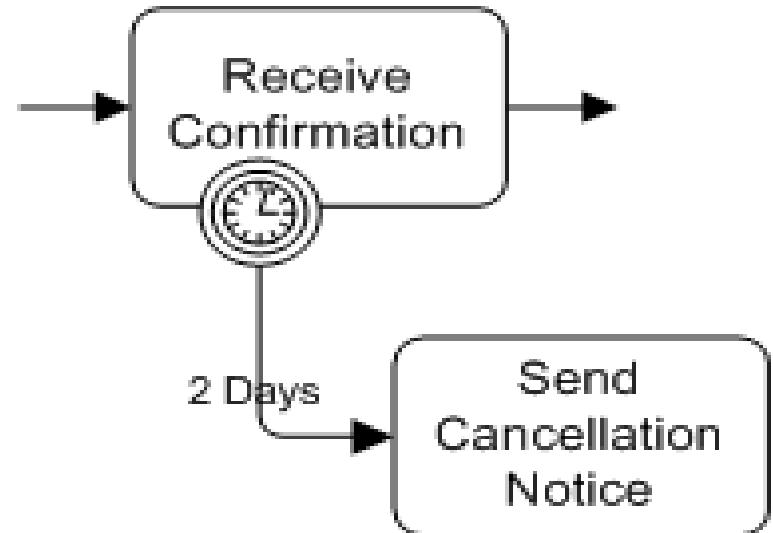
Beispiel: aus dem B2B-Bereich mit 2 verschiedenen Pools und 2 Lanes.

B2B Modeling (nicht nur Unternehmens- sondern auch Abteilungs-/Personen-übergreifend)



Beispiel: Urlaubsantrag eines Mitarbeiters

Exception Handling and Timer

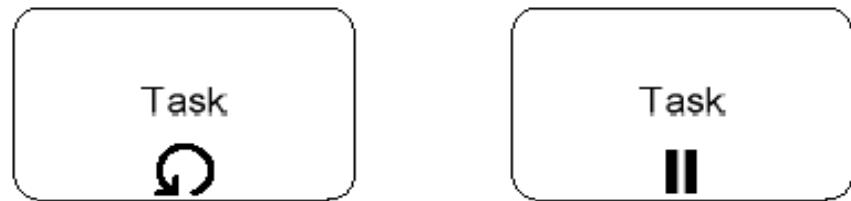


- Ereignisse für eine Unterbrechung (Intermediate Events) an einer Grenze einer Aktivität repräsentieren Trigger, die die Aktivität unterbrechen können.
- Alle Arbeitsaufgaben innerhalb der Aktivität werden beendet und der Prozess von diesem Ereignis aus weitergeführt.
- Timer, Exceptions, Messages, etc. können mögliche Trigger sein.

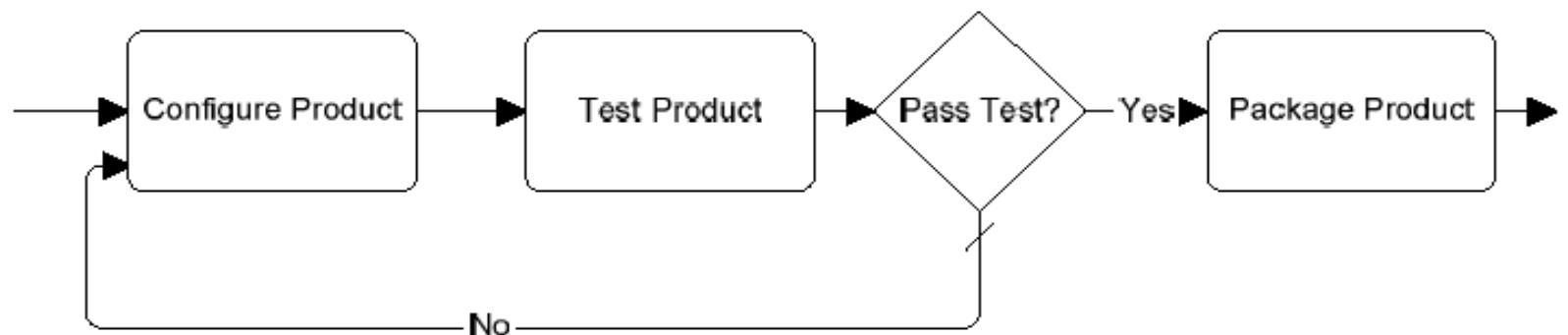


Beispiel: Timer, um einen Prozess-Fluss zu verzögern.

Looping (Schleifen)

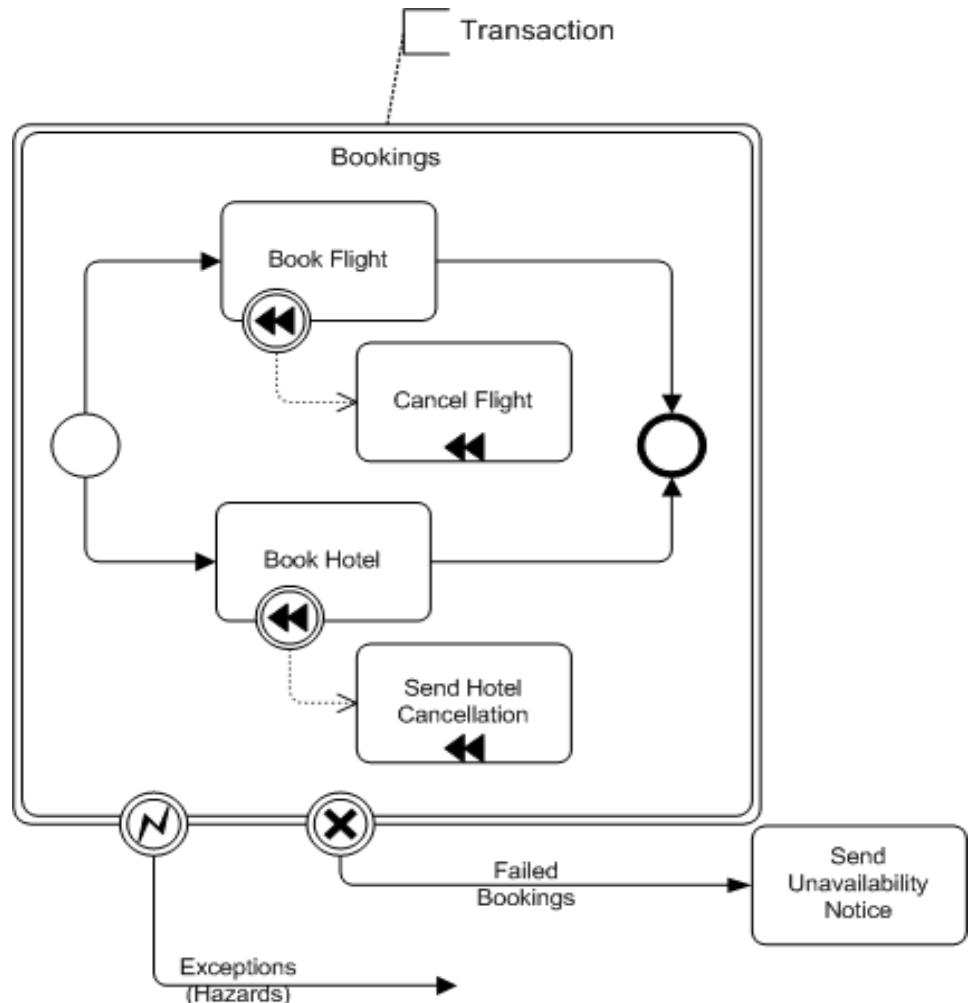


Activity Looping: Do-While;
While-Do; Multiple Instance



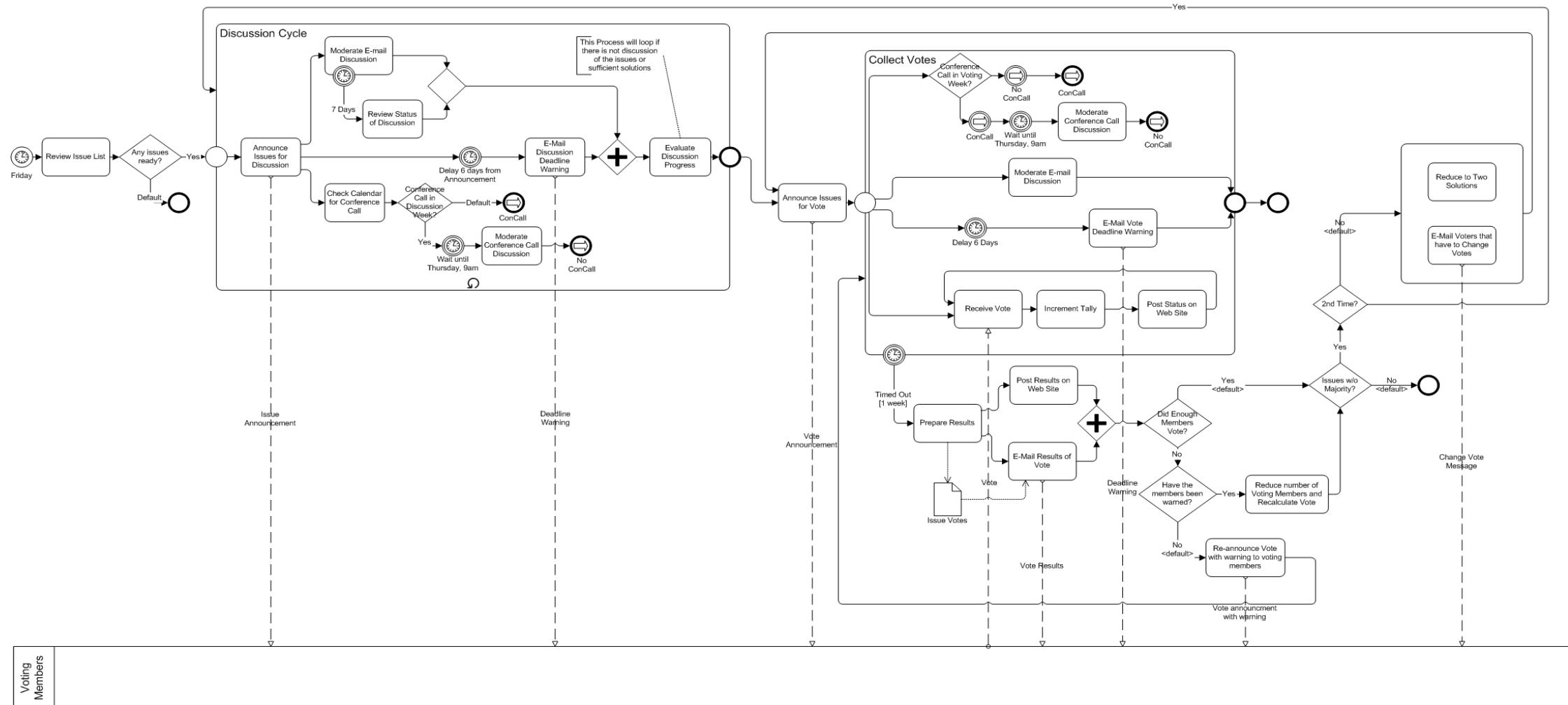
Sequence Flow Looping

Compensation Handling and Transactions

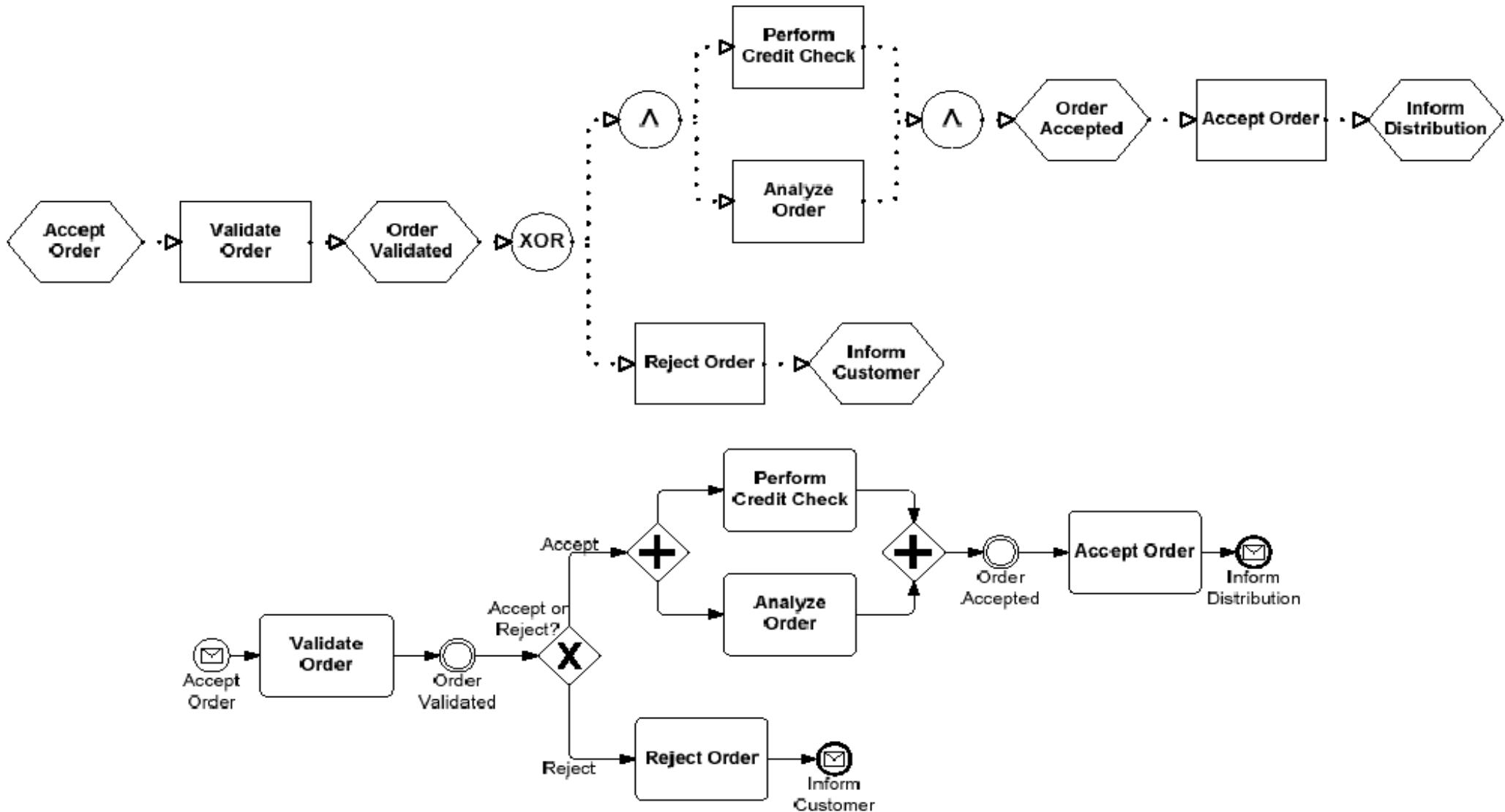


- Eine Transaction ist eine Aktivität, die eine "doppelte" Grenze hat. Transactions werden von einem transaction protocol unterstützt (z. B., WS-Transaction).
- Eine normale, ausgehende Sequenz steht für einen Pfad einer erfolgreichen Durchführung.
- Ein Ereignis-Abbruch (Cancel Intermediate Event) steht für einen Pfad der abgebrochenen Durchführung.
- Ein Ausnahme-Ereignis (Exception Intermediate Event) steht z.B. für einen Prozess-Pfad im Sinne einer Gefährdung / eines Risikos der Durchführung.
- Aktivitäten, die für Ersatzbehandlungen (compensation) stehen, verlaufen außerhalb des normalen Flusses, sind aber mit normalen Aktivitäten verbunden.

A Complex Process



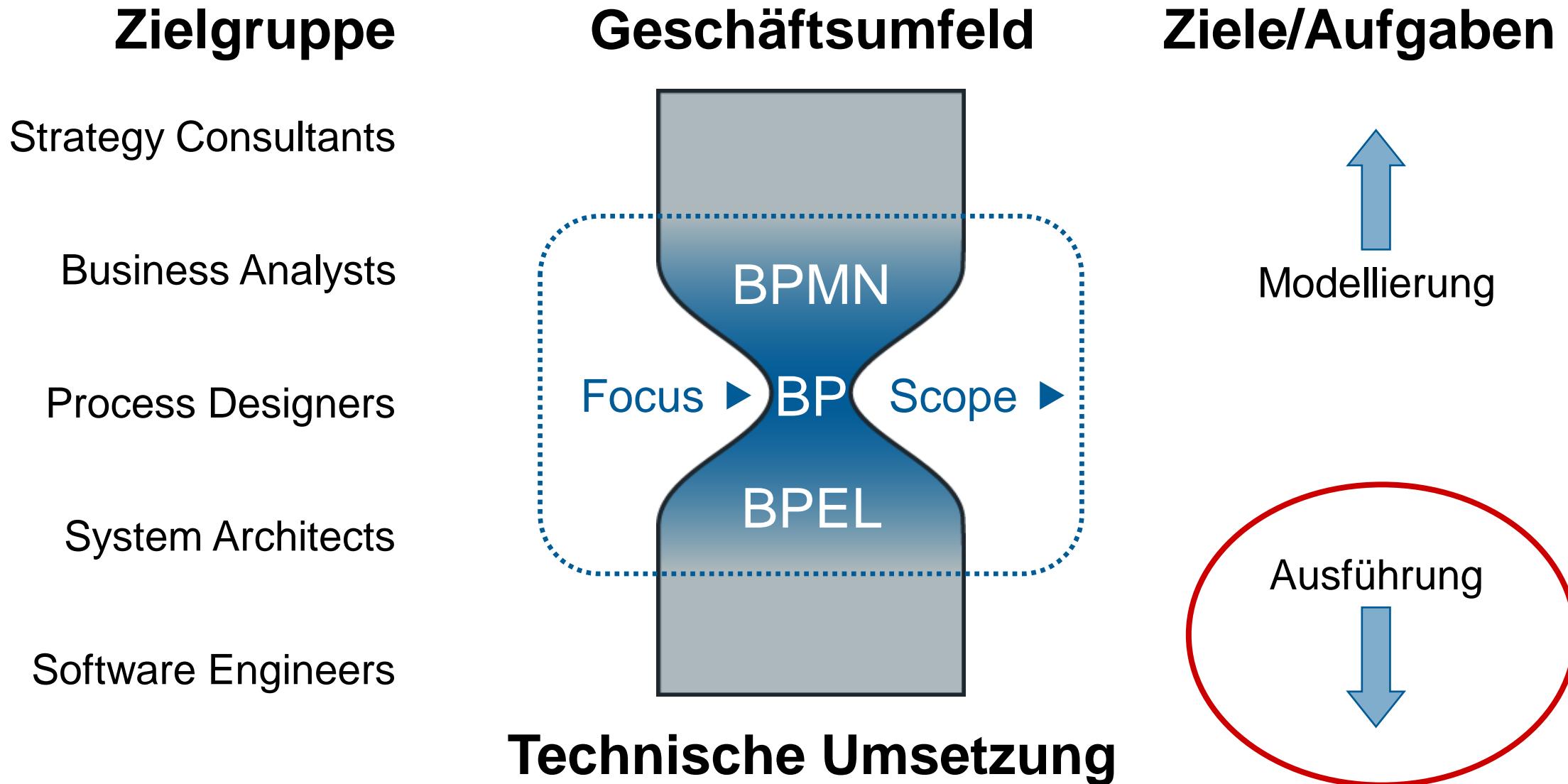
Beispiel: EPK und BPMN



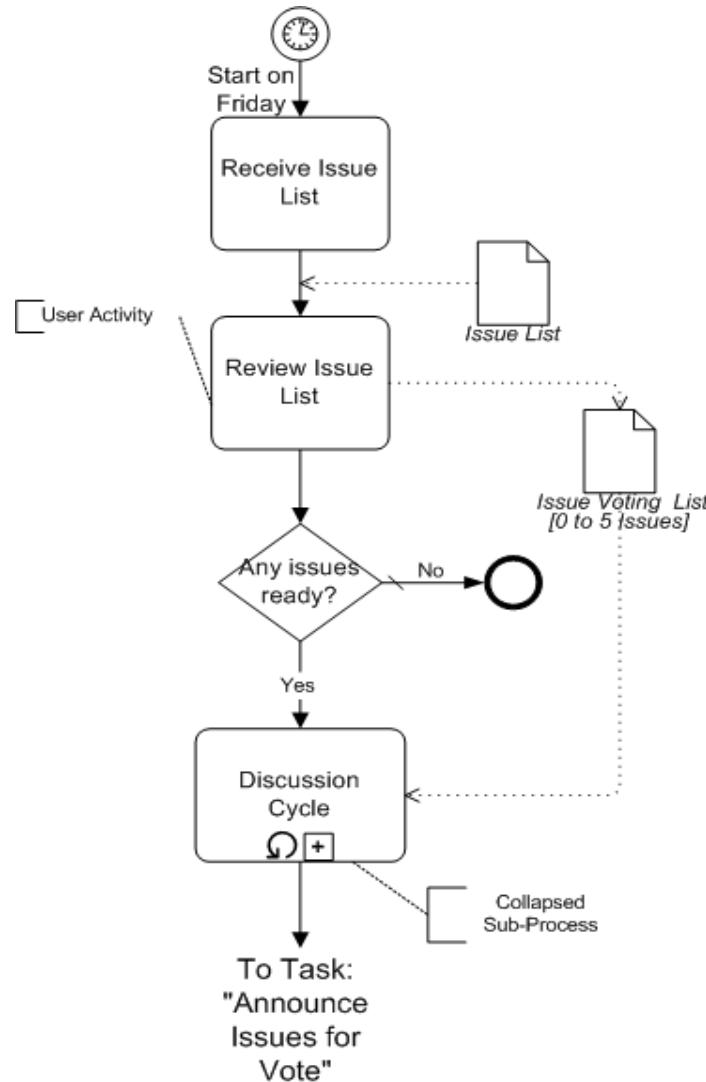
Notation

- Business Process Diagram Elements
 - Core Set of Diagram Elements
 - Complete Set of Diagram Elements
- Business Process Diagram Samples
 - Normal Flow
 - B2B Modeling
 - Exception Handling
 - Compensation Handling
 - A Complex Process
- Mapping to WS-BPEL Sample (ehemals BPEL4WS)

BPM-„Sanduhr“: Geschäftsprozesse als Schnittpunkt zwischen Modellierung und Ausführung



Mapping to WS-BPEL Sample (BPEL4WS)



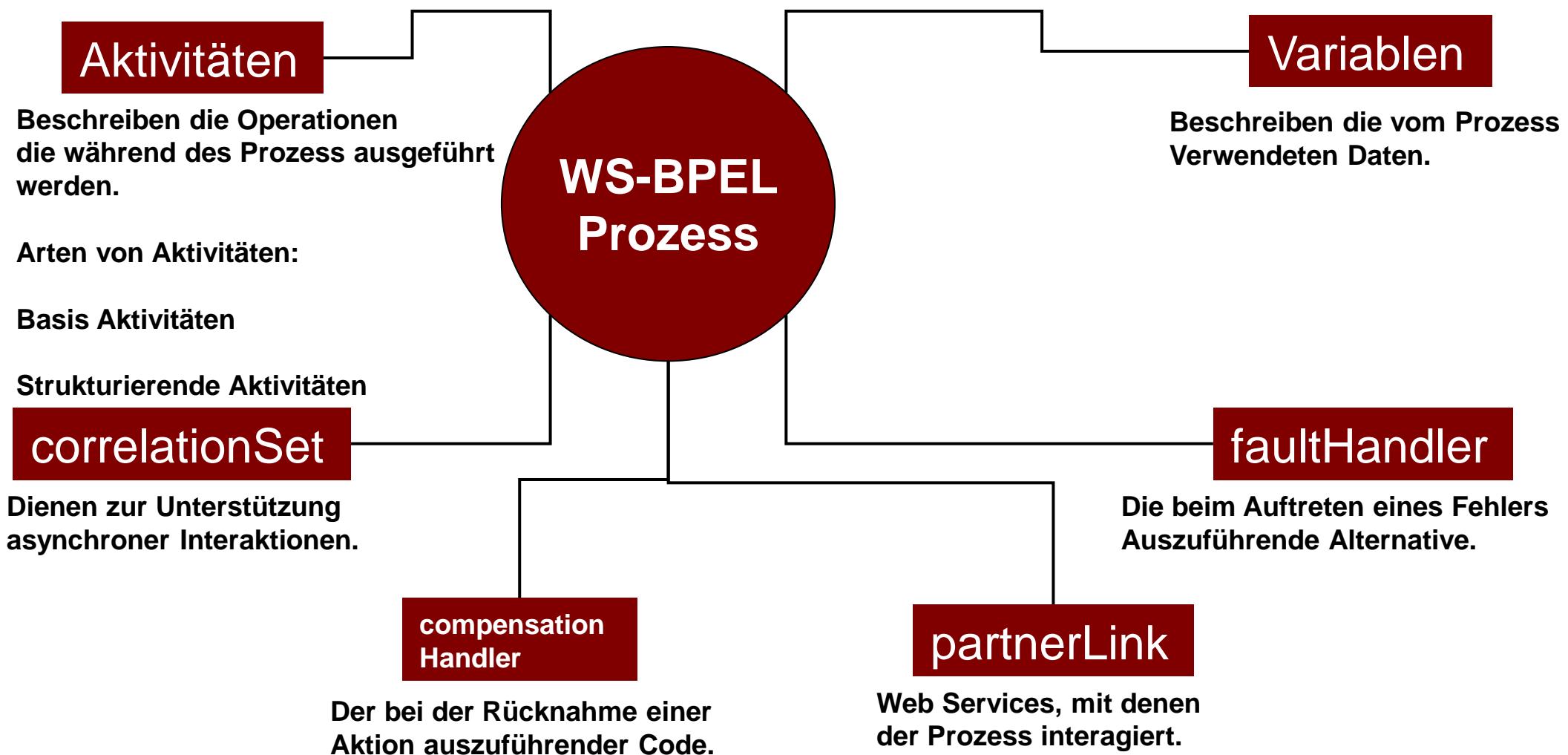
```
<process name="EMailVotingProcess">
  <!-- The Process data is defined first-->
  <sequence>
    <receive partnerLink="Internal" portType="tns:processPort"
            operation="receiveIssueList" variable="processData"
            createInstance="Yes"/>
    <invoke name="ReviewIssueList" partnerLink="Internal"
            portType="tns:internalPort" operation="sendIssueList"
            inputVariable="processData" outputVariable="processData"/>
    <switch name="Anyissuesready">
      <!-- name="Yes" -->
      <case condition="bpws:getVariableProperty(ProcessData,NumIssues)>0">
        <invoke name="DiscussionCycle" partnerLink="Internal"
                portType="tns:processPort" operation="callDiscussionCycle"
                inputVariable="processData"/>
        <!-- Other Activities not shown -->
        <!--name="No" -->
      </case>
      <otherwise>
        <empty/>
      </otherwise>
    </switch>
  </sequence>
</process>
```

- Was ist WS-BPEL (Business Process Execution Language for WebsServices)?
 - Ist eine Entwicklung von Microsoft, IBM und BEA.
 - WS-BPEL ist XML-basiert und eröffnet die Möglichkeit der Interaktionen von Geschäftsprozessen sowohl innerhalb als auch zwischen Unternehmen.
 - Definiert dafür ein Modell und eine Grammatik, zur Beschreibung des Verhaltens eines Geschäftsprozesses basierend auf Interaktionen zwischen dem Prozess und seinen Partnern (WS).

Business Process Execution Language for Web-Services (Eigenschaften von WS-BPEL)

- Zwei Arten von Geschäftsprozessen:
 - **Ausführbaren Geschäftsprozessen** (*executable business process*)
 - Sie modellieren das tatsächliche Verhalten der Teilnehmer einer Geschäftsbeziehung.
 - **Geschäftsprotokolle** (*business protocols*)
 - Sie verwenden abstrakte Prozesse zur Beschreibung des öffentlichen Verhaltens der Parteien beim Nachrichtenaustausch.

Kernkomponenten von WS-BPEL



```
<flow> ← Führt alle Aktivitäten parallel aus.  
<links> ← Definiert eine Abhängigkeit zwischen einer Quell-  
und Zielaktivität.  
    <link name="order-to-airline"/>  
    <link name="airline-to-agent"/>  
</links>  
<receive partner="customer" portType="itineraryPT"  
        operation="sendItinerary" container="itinerary">  
    <source linkName="order-to-airline"/>  
</receive> ← Der Prozess erhält eine Nachricht von einem  
Partner.  
<invoke partner="airline" portType="ticketOrderPT"  
        operation="requestTickets" inputContainer="itinerary">  
    <target linkName="order-to-airline"/>  
    <source linkName="airline-to-agent"/>  
</invoke> ← Der Prozess ruft eine Operation bei einem Partner auf.  
....
```

Geschäftsprozess-Management

Prof. Dr.-Ing. Andreas Ittner

Email: ittner@hs-mittweida.de

WWW: www.hs-mittweida.de/~ittner

Tel.: +49(0)3727-58-1288

Mob.: +49(0)177-5555-347

Fax.: +49(0)180-35518-22467

Gliederung

- Motivation
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs, BPMN, ...
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards

1. Fragestellungen und klassische Fehler,
2. Struktur-Elemente und -Eigenschaften (Zusammenhängend, Free Choice, Workflow-Netz, well-handled, -structured, S-Komponenten / s-coverable),
3. Dynamische Elemente und Eigenschaften (Anfangsmarkierung, Beschränktheit, Komplementbildung, tot, lebendig, verklemmungsfrei, Soundness, Invarianten (S-, T-Invarianten)),
4. Erreichbarkeitsanalyse,
5. Linear Algebraische Darstellung,
6. Zusammenhänge bei der Prozess-Analyse.

1. Fragestellungen und klassische Fehler,
2. Struktur-Elemente und -Eigenschaften (Zusammenhängend, Free Choice, Workflow-Netz, well-handled, -structured, S-Komponenten / s-coverable),
3. Dynamische Elemente und Eigenschaften (Anfangsmarkierung, Beschränktheit, Komplementbildung, tot, lebendig, verklemmungsfrei, Soundness, Invarianten (S-, T-Invarianten)),
4. Erreichbarkeitsanalyse,
5. Linear Algebraische Darstellung,
6. Zusammenhänge bei der Prozess-Analyse.

Qualitative Fragestellungen

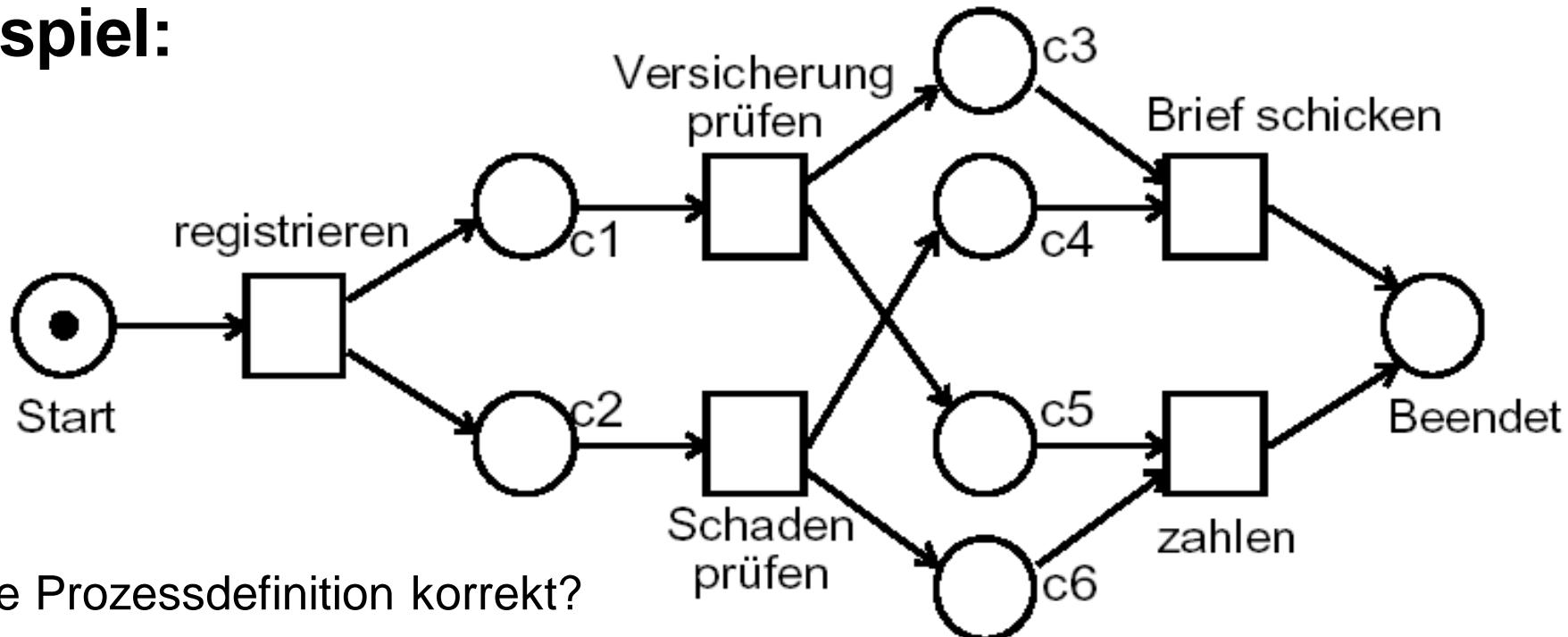
- Kann es zu Deadlocks (toten Markierungen) kommen?
- Können alle Arten von Cases (Fällen) erfolgreich behandelt werden?
- Werden alle Cases irgendwann beendet?
- Können Tasks (Aufgaben) parallel ausgeführt werden?

Quantitative Fragestellungen

- Wie viele Cases pro Stunde können bearbeitet werden?
- Was ist die durchschnittliche Durchlaufzeit?
- Wie viele Ressourcen werden pro Case benötigt?
- Was ist die durchschnittliche Wartezeit?

1. Fragestellungen und klassische Fehler

Beispiel:

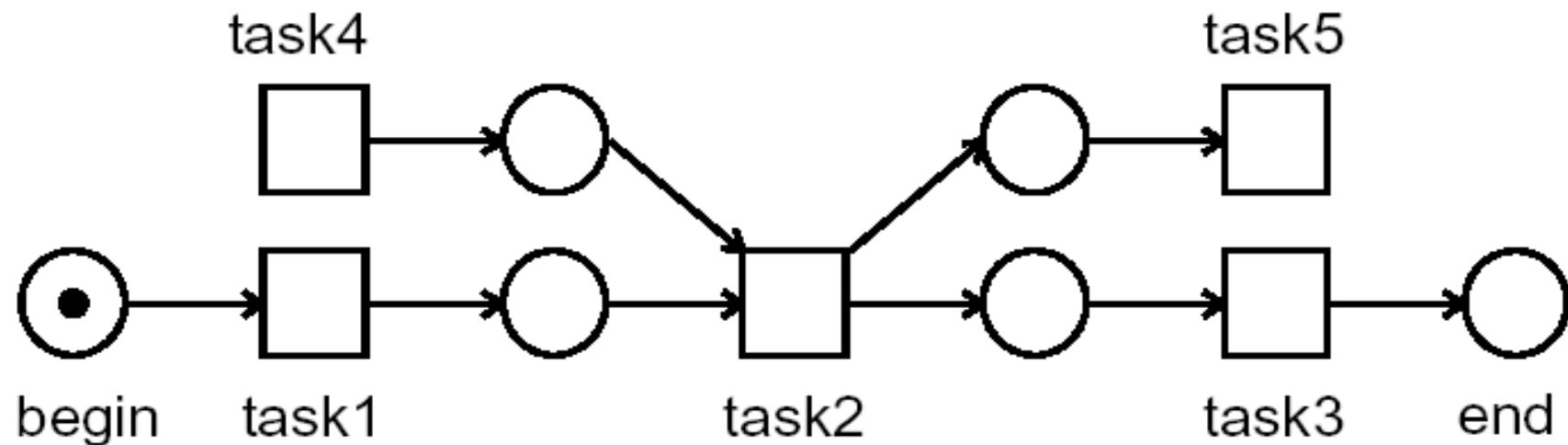


Ist die Prozessdefinition korrekt?

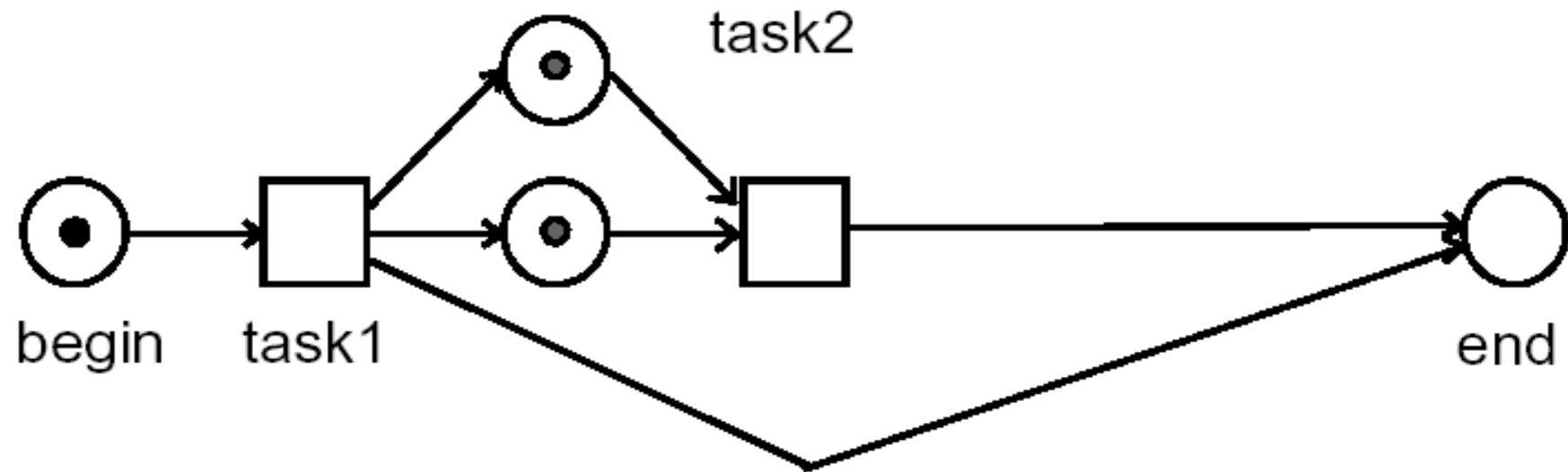
Gibt es tote Transitionen oder Deadlocks?

Ist das Netz k-beschränkt?

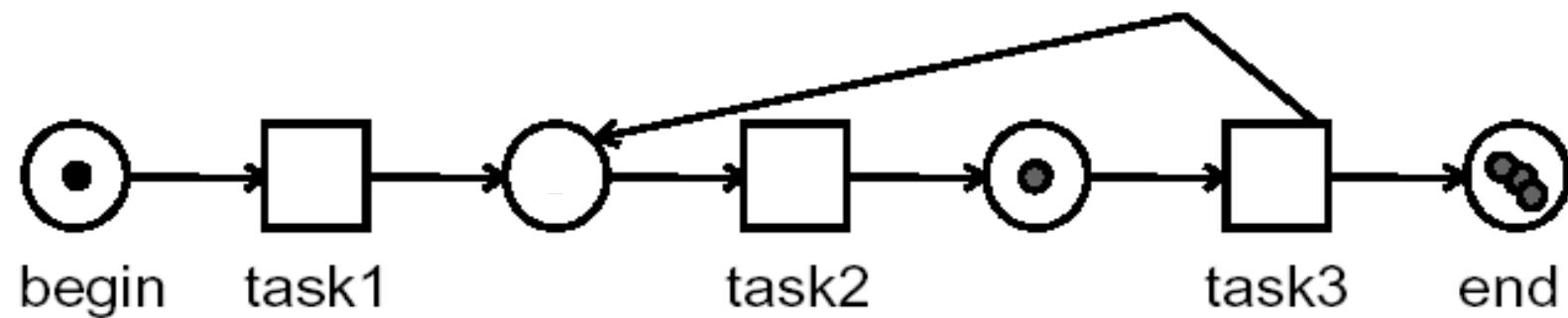
Beispiel: „baumelnde“ Tasks:



Beispiel: mögliche tote Markierungen (Deadlocks):



Beispiel: Unbeschränkt und unendlich:



2. Struktur-Elemente und -Eigenschaften

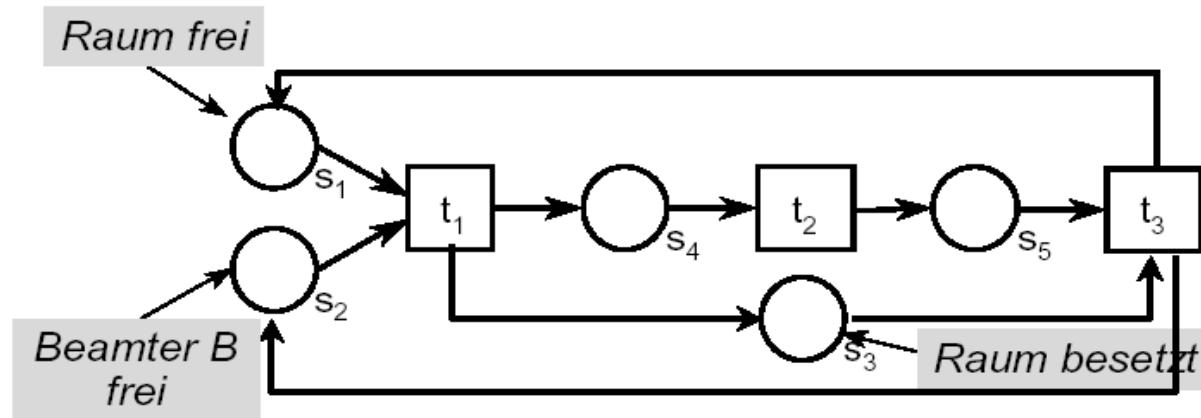
- Eine **Struktureigenschaft** eines Petrinetzes hängt nicht von der Markierung des Netzes ab. Es werden Aussagen über das Verhalten aufgrund der Struktur getroffen.
- Stellen und Transitionen (Knoten oder Netzelemente), Kanten wobei **Stellen** passive Komponenten und **Transitionen** aktive Komponenten darstellen.
- Ein Petrinetz beschreibt immer gewisse Struktur-Beziehungen zwischen den Komponenten eines Systems,
 - Vorbereich, Nachbereich,
 - Vorwärtsverzweigung, Rückwärtsverzweigung,
 - Teilnetz, Rand (stellen-, transitionsberandet),
 - Hierarchieerweiterung (in der Entwurfsphase eingesetzt, Darstellung unterschiedlicher Sichtweisen).

2. Struktur-Elemente und -Eigenschaften



Beispiel: Schalterraum einer Behörde

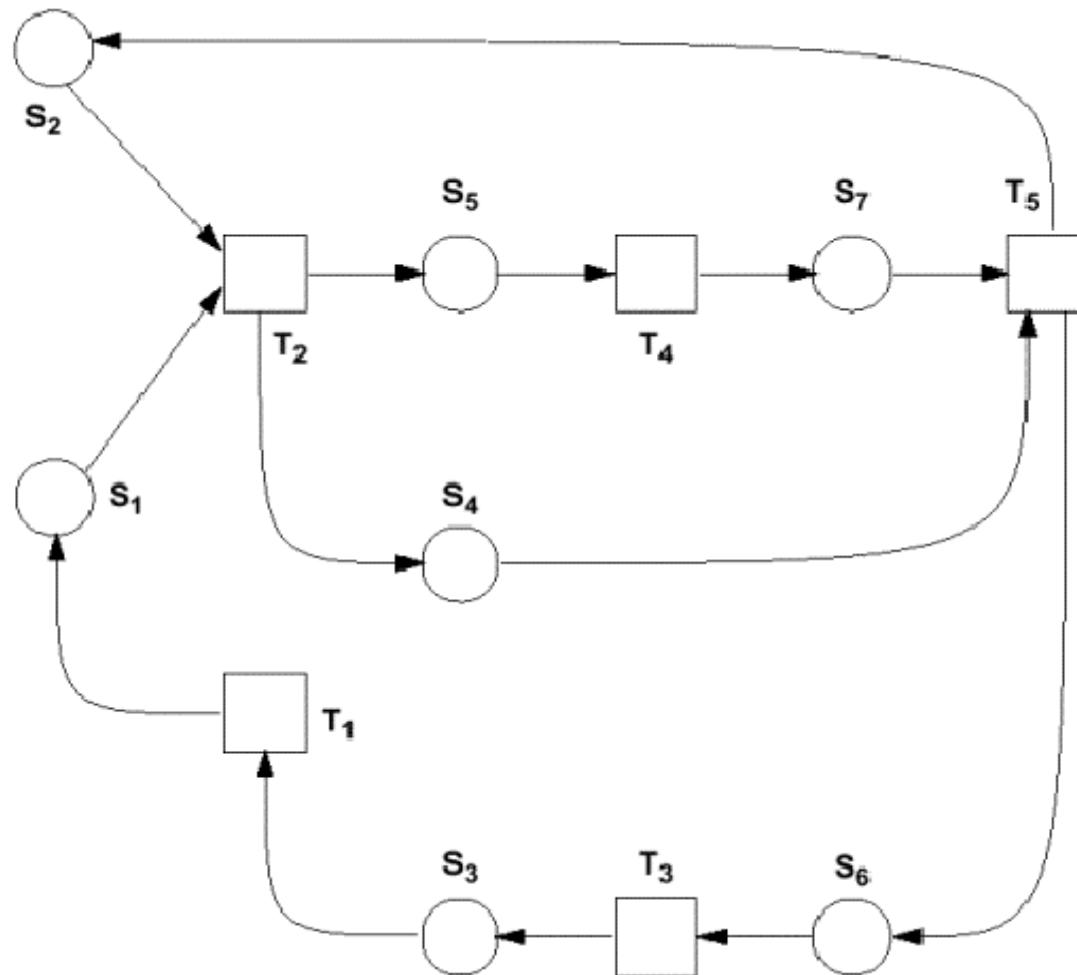
- Ein Schalterraum ist mit dem Beamten B besetzt. Er ist für die Bearbeitung eines Antrages zuständig. Dies erfolgt in Anwesenheit des Antragstellers A. Aus Datenschutzgründen darf nur ein Antragsteller im Raum sein.
- mögliche Petrinetz-Modellierung:



- Für eine bestimmte Modellierung kann durch Beschriftung ein Anwendungsbezug hergestellt werden.

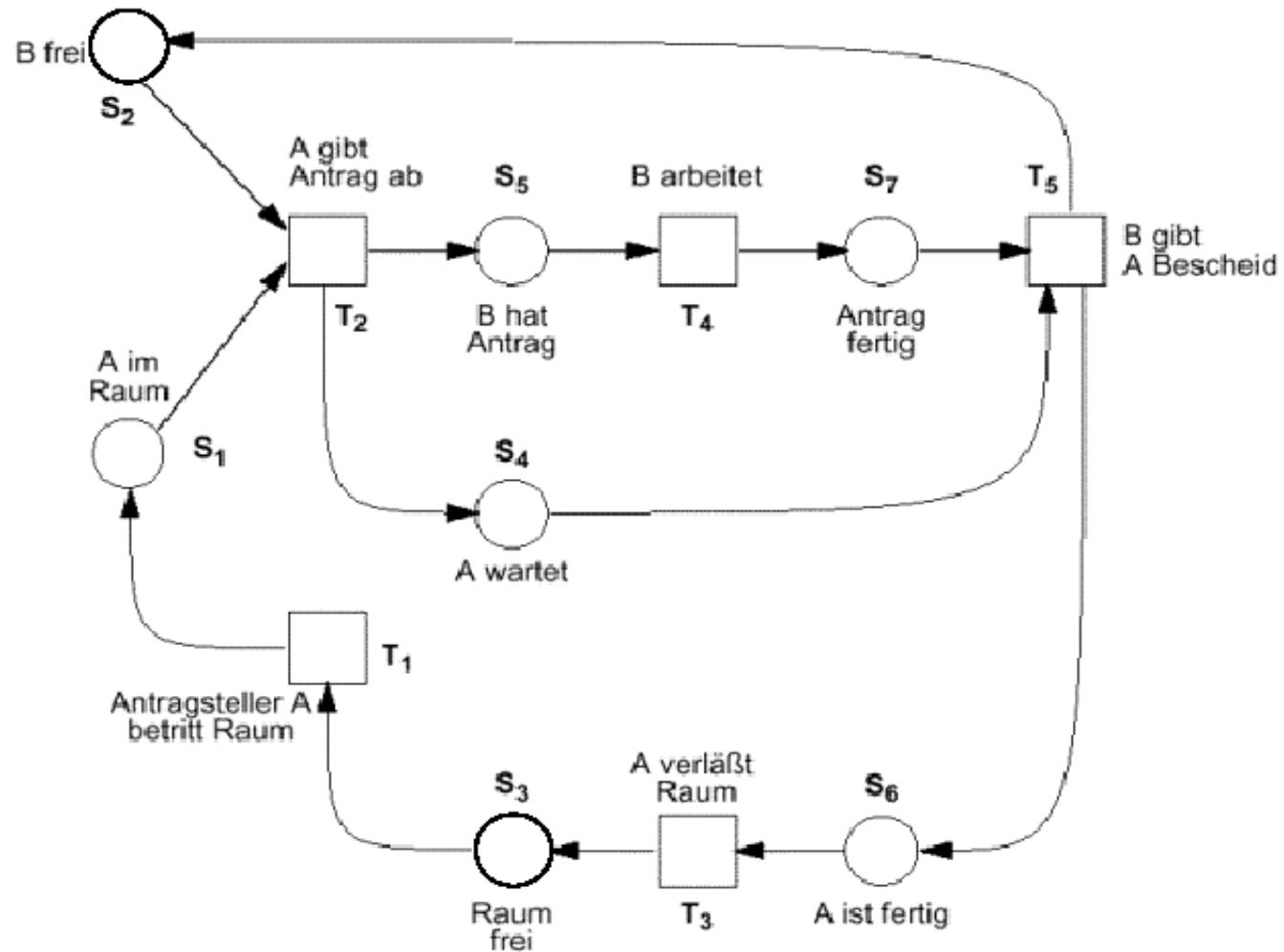
2. Struktur-Elemente und -Eigenschaften

andere mögliche Petrinetz-Modellierung:



2. Struktur-Elemente und -Eigenschaften

Herstellung des Anwendungsbezuges:



2. Struktur-Elemente und -Eigenschaften

Ein Netz $N = (S, T, F)$ ist ***zusammenhängend***,
wenn keine Zerlegung in X_1 und X_2 ($S \cup T = X_1 \cup X_2$), mit
 $X_1, X_2 \neq \emptyset$,
 $X_1 \cap X_2 = \emptyset$ mit $F \subseteq (X_1 \times X_1) \cup (X_2 \times X_2)$ existiert.

Es darf also kein Teilnetz existieren, das keine Verbindung zum Rest des Netzes hat.

N heißt ***stark zusammenhängend***,
wenn für je zwei Elemente $x, y \in S \cup T$ mit $x \neq y$
eine Sequenz $(z_1, z_2), (z_2, z_3), \dots, (z_{n-1}, z_n) \in F$ existiert ($n \geq 2$),
so dass $x = z_1$ und $y = z_n$.

Von jedem Knoten besteht also eine gerichtete Verbindung zu jedem beliebigen Knoten.

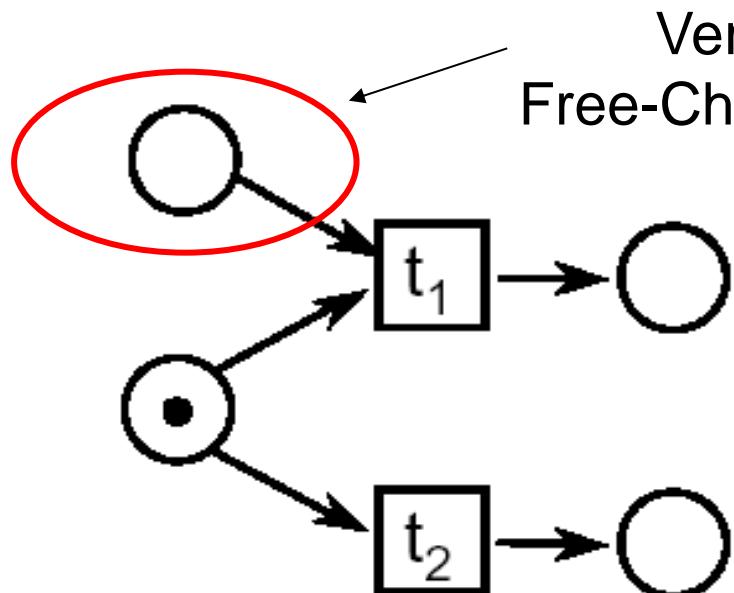
2. Struktur-Elemente und -Eigenschaften

- Ein **Free-Choice-Netz** ist ein Netz bei dem
 1. die Transitionen einer vorwärts verzweigten Stelle nicht rückwärts verzweigt sind, bzw.
 2. die Stellen einer rückwärts verzweigten Transition nicht vorwärts verzweigt sind.
- bei einem Konflikt kann also zwischen den Transitionen frei und unabhängig von anderen Stellen ausgewählt werden.
- bei einer Synchronisation werden die Marken in einer Transition wieder zusammengeführt unabhängig von anderen Transitionen.
- Motiv:
 - strukturierte Modellierung,
 - bessere Analysierbarkeit,
 - Modellierung von Geschäftsprozessen (GP) möglich.

2. Struktur-Elemente und -Eigenschaften

- **Konflikt:**

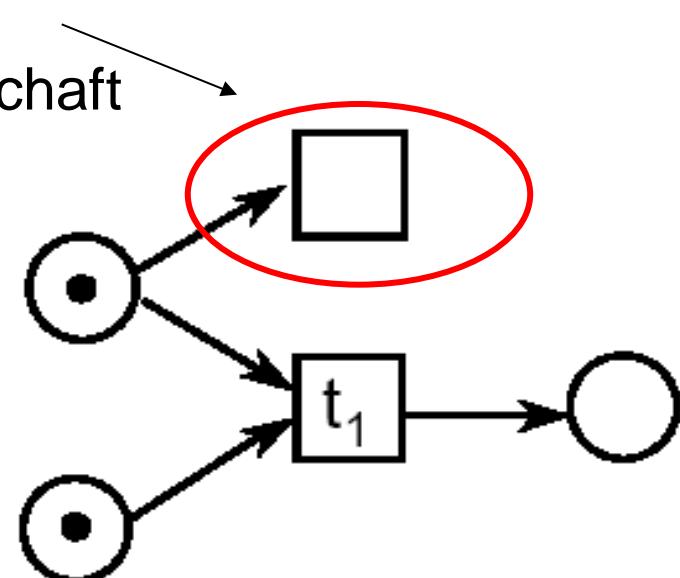
- Zwei Transitionen benötigen die gleiche Marke(n)



Die Transitionen einer vorwärts verzweigten Stelle dürfen nicht rückwärts verzweigt sein!

- **Synchronisation:**

- Transition kann erst schalten, wenn alle Eingangsstellen markiert sind.



Die Stellen einer rückwärts verzweigten Transition dürfen nicht vorwärts verzweigt sein!

2. Struktur-Elemente und -Eigenschaften

- Für ein **Free-Choice-Netz** gilt:

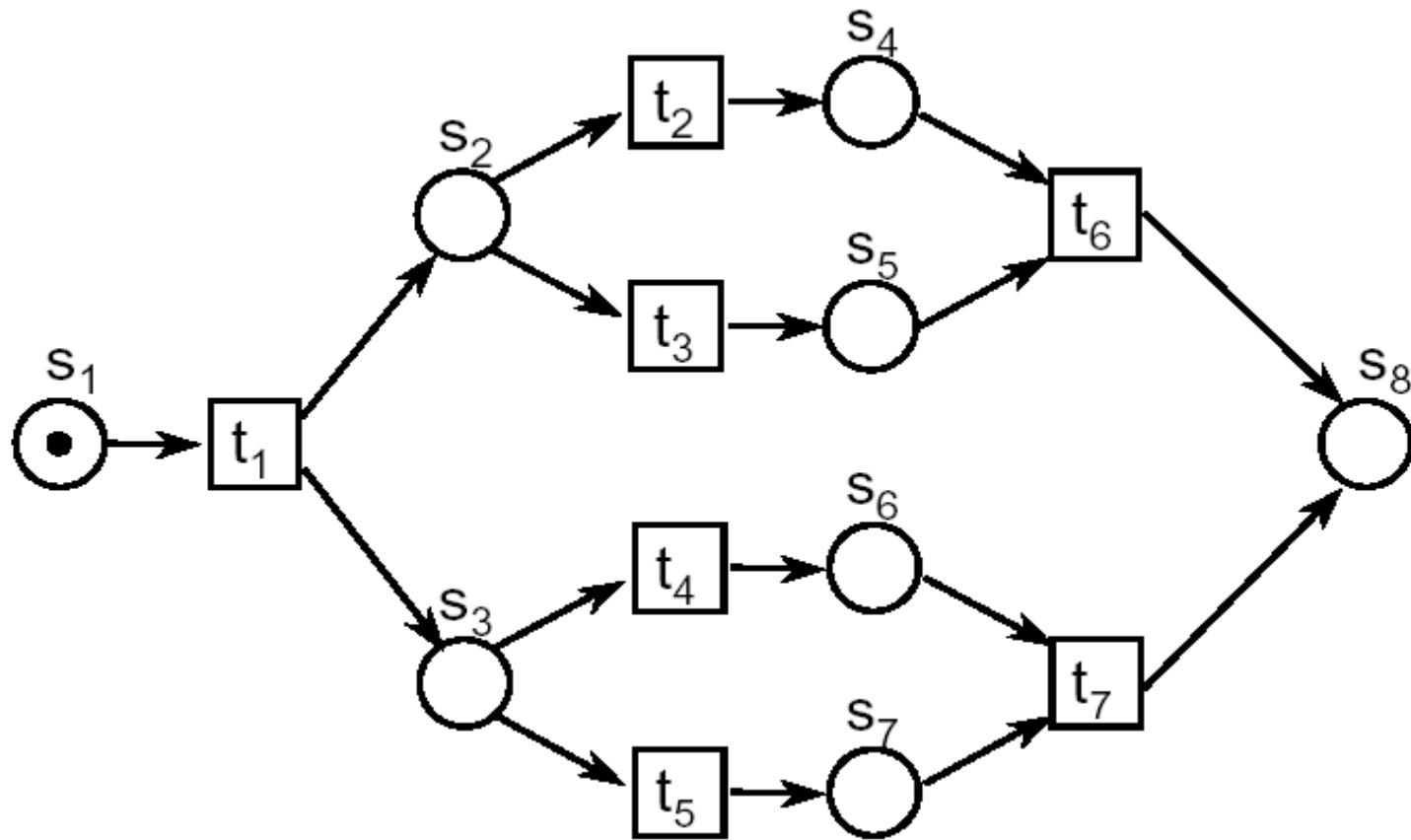
- für Transitionen:

$$\forall t_1, t_2 \in T: \quad \bullet t_1 \cap \bullet t_2 \neq \emptyset \quad \Rightarrow \quad \bullet t_1 = \bullet t_2 = \{ s \}$$

- für Stellen:

$$\forall s_1, s_2 \in S: \quad s_1 \bullet \cap s_2 \bullet \neq \emptyset \quad \Rightarrow \quad s_1 \bullet = s_2 \bullet = \{ t \}$$

Beispiel: ein Free-Choice-Netz



2. Struktur-Elemente und -Eigenschaften

- Welche Struktureigenschaften hat ein korrektes **Workflow**-Prozess-Modell?
 - definierter Anfang und definiertes Ende („davor“ und „danach“ ist irrelevant),
 - keine Aufgaben, die nie ausgeführt werden,
 - keine Aufgaben, die nicht zum Ende führen,
- Schlussfolgerung für das Petrinetz:
 - eine Stelle, die die Erzeugung des Falles darstellt (Start),
 - diese Stelle hat keinen Input,
 - eine Stelle, die die Beendigung des Falles darstellt (Ende),
 - diese Stelle hat keinen Output,
 - für jede Transition gilt:
 - es gibt einen Pfad von der Start-Stelle und
 - es gibt einen Pfad zur Ende-Stelle.

→ ein Petri-Netz, das diese Eigenschaften erfüllt, bezeichnen wir als **Workflow-Netz** (WF-Netz).

2. Struktur-Elemente und -Eigenschaften

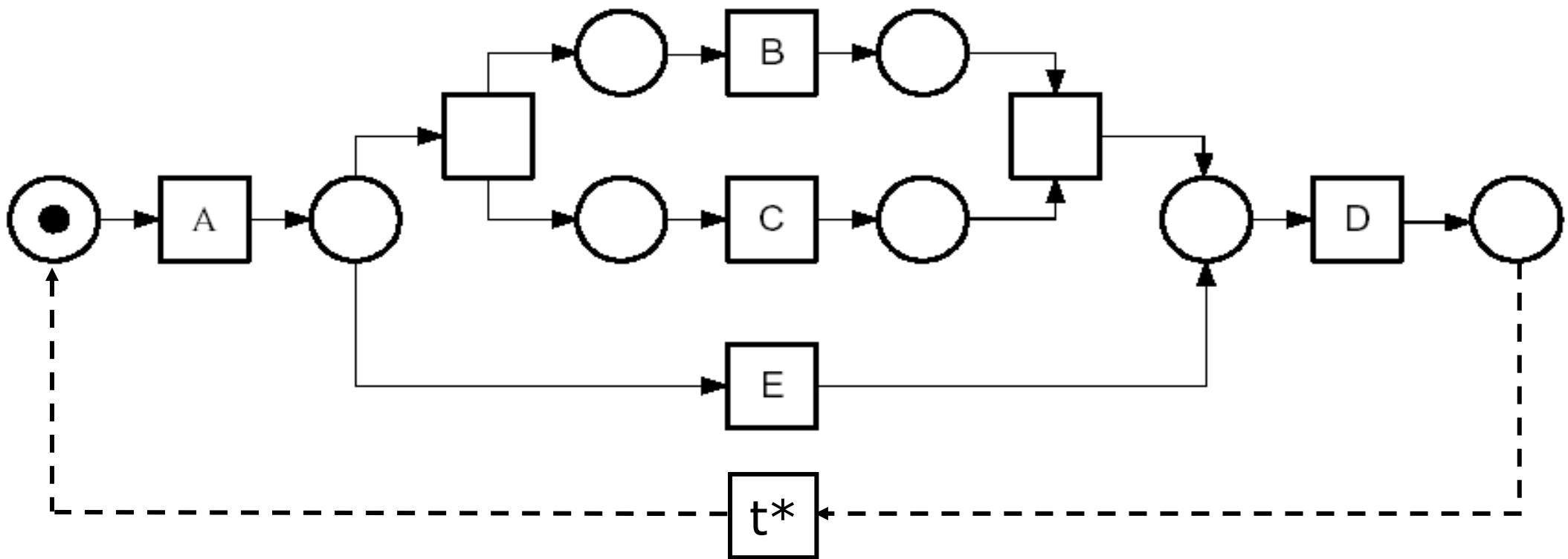
Ein **Workflow-Netz** ist ein Petrinetz $N = (S, T, F)$, für welches gilt:

- N hat zwei Stellen i und o mit den folgenden Eigenschaften:
 - Stelle i ist eine Quelle, d.h. $\bullet i = \emptyset$,
 - Stelle o ist eine Senke, d.h. $o \bullet = \emptyset$.
- Wenn zu N eine Transition t^* hinzugefügt wird, die o und i verbindet mit $\bullet t^* = \{o\}$, $t^* \bullet = \{i\}$, so ist das resultierende Netz stark zusammenhängend.
- Folgerung: i und o sind eindeutig.
- Ein WF-Netz ist das Petrinetz-Modell eines Workflow-Prozesses.

Bemerkung: Die Stellen i und o sind eindeutig und Teil des WF-Netzes, während Transition t^* nicht zum WF-Netz gehört, sondern nur zur Prüfung dient, ob das Netz stark zusammenhängend ist.

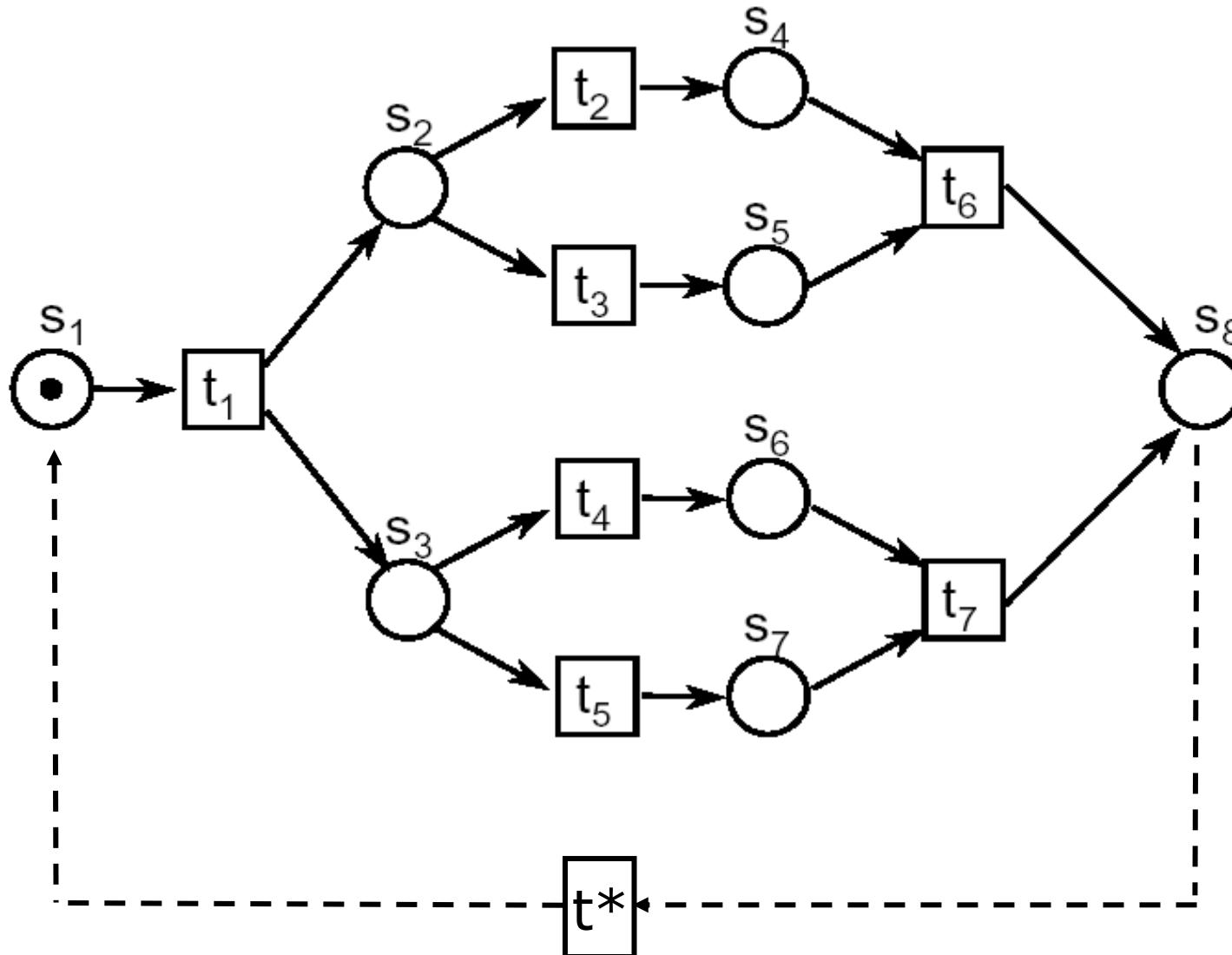
2. Struktur-Elemente und -Eigenschaften

Beispiel: Workflow-Netz (nicht stark zusammenhängend, erst durch Einfügung einer Transition t^* (als Prüfung) wird es stark zusammenhängend)



2. Struktur-Elemente und -Eigenschaften

Frage: ist dieses Free-Choice-Netz auch ein WF-Netz?

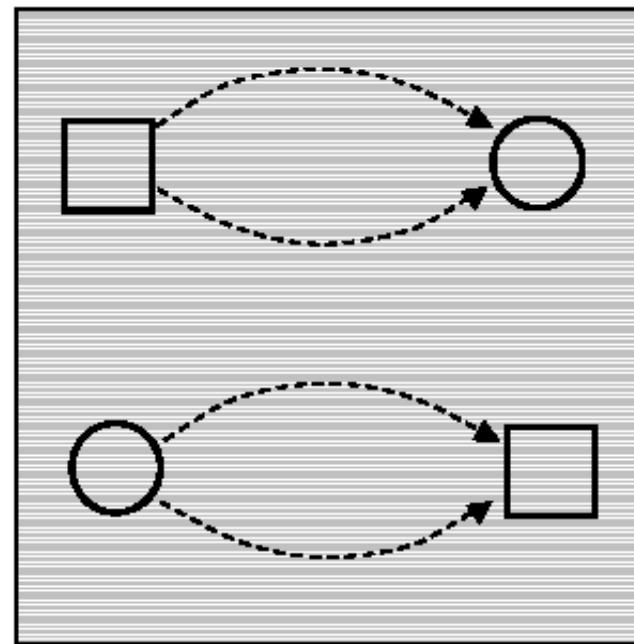


2. Struktur-Elemente und -Eigenschaften

Routing-Kombinationen:



„gute“ Routingkonstrukte



„schlechte“ Routingkonstrukte

AND-splits mit AND-joins und OR-splits mit OR-joins vereinigen

2. Struktur-Elemente und -Eigenschaften

Well-handled, well-structured

Ein WF-Netz $N = (S, T, F)$ ist genau dann ***well-handled***,

wenn für je zwei Knoten x, y

($x, y: x \in T$ und $y \in S$ oder $x \in S$ und $y \in T$)

keine zwei Pfade von x nach y existieren, die

- aus mehr als 2 Elementen bestehen UND

- nur x, y gemeinsam haben.

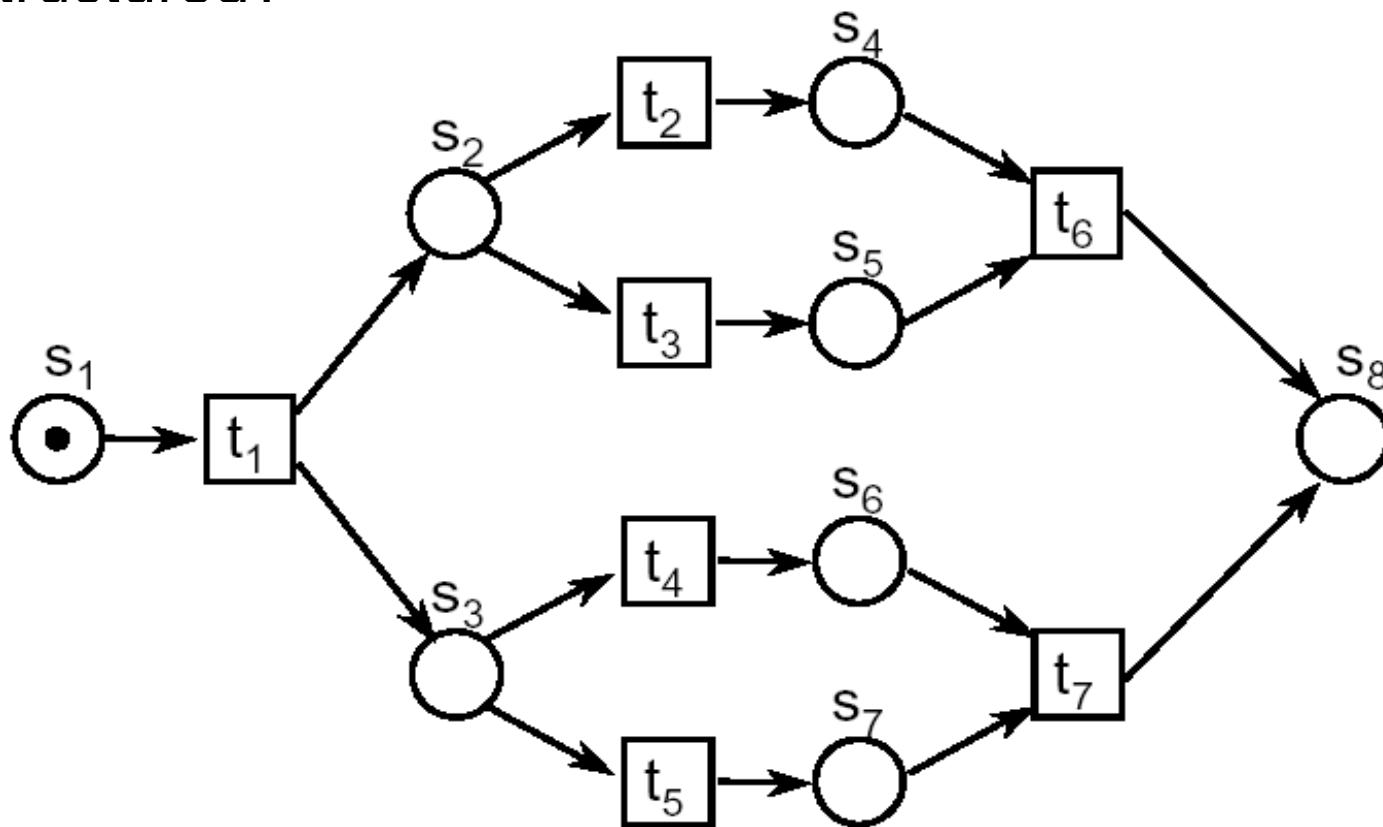
Ein WF-Netz ist ***well-structured***,

wenn das um t^* (mit $\bullet t^* = \{o\}$, $t^{*\bullet} = \{i\}$)

erweiterte Netz well-handled ist.

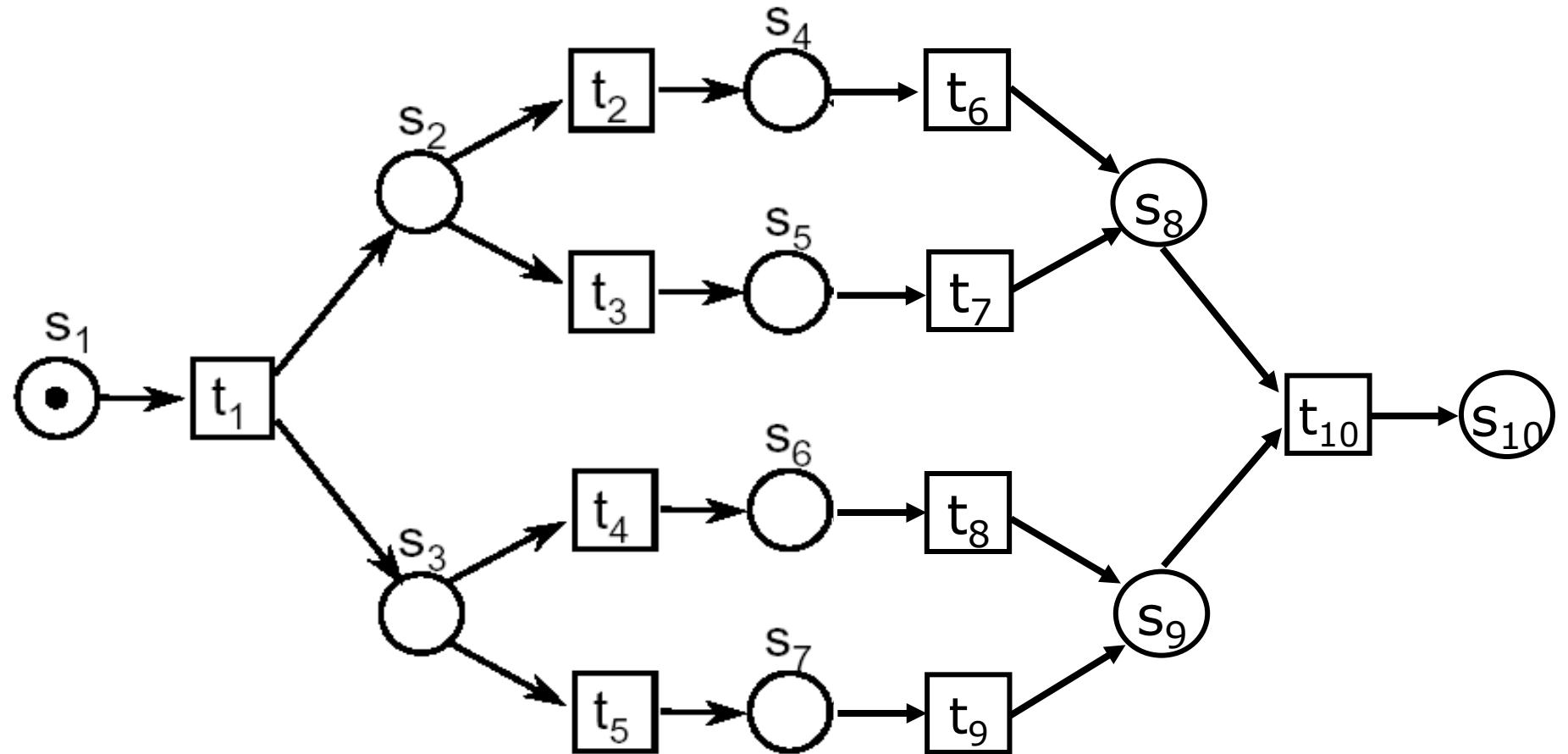
2. Struktur-Elemente und -Eigenschaften

Frage: ist dieses Free-Choice-Netz auch well-handled, vielleicht sogar well-structured?



→ Nein, wegen (t_1, s_8) , (s_2, t_6) und (s_3, t_7) !

Beispiel: für ein well-handled und well-structured Netz

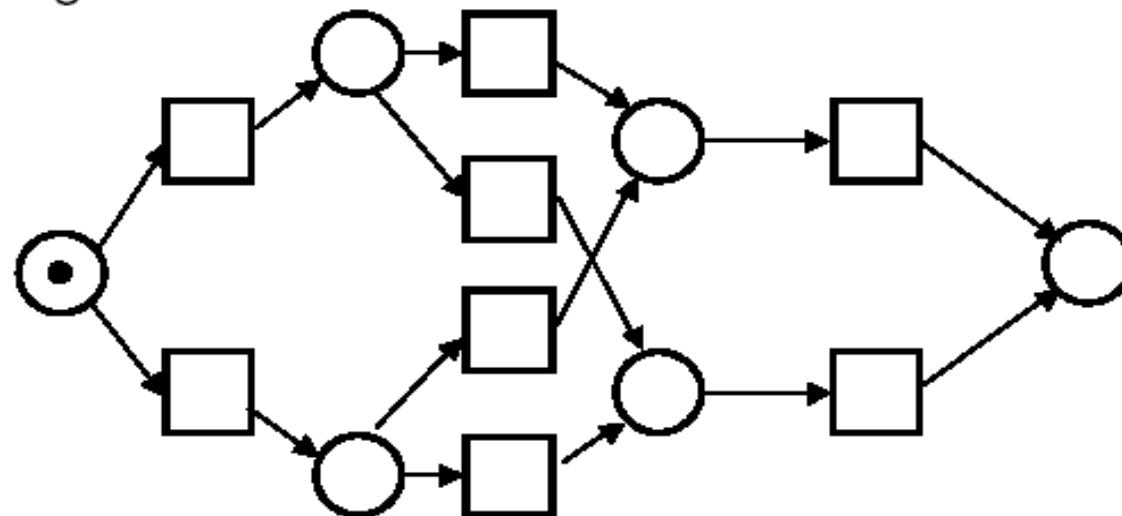


2. Struktur-Elemente und -Eigenschaften

Ein WF-Netz $N = (S, T, F)$ ist genau dann ein **Zustandsautomat**, wenn gilt:

$$\forall t \in T: |\bullet t| = |t\bullet| = 1,$$

d.h. jede Transition in $t \in T$ hat genau eine Eingangs- und genau eine Ausgangskante.



Motiv: - immer nur genau eine Marke im System,
Bearbeitung eines Falles kann also nur sequenziell
ablaufen

Bemerkung: Es werden weder Marken erzeugt, noch verschwinden welche!
Modellierung paralleler Abläufe nicht möglich!

2. Struktur-Elemente und -Eigenschaften

Ein Teilnetz $N' = (S', T', F')$ eines WF-Netzes $N = (S, T, F)$ mit $S' \subseteq S, T' \subseteq T, F' \subseteq F$, ist eine **S-Komponente** von N genau dann, wenn:

- N' ist stark zusammenhängend,
- N' ist Zustandsautomat,
- $\forall s \in S': \quad \bullet s \cup s \bullet \subseteq T'$.

Ein WF-Netz $N = (S, T, F)$ ist **s-coverable** genau dann, wenn $\forall s \in S \quad \exists \text{ S-Komponente } N' = (S', T', F') \text{ von } N: s \in S'$.

Für jede Stelle $q \in S'$ und $t \in T$:

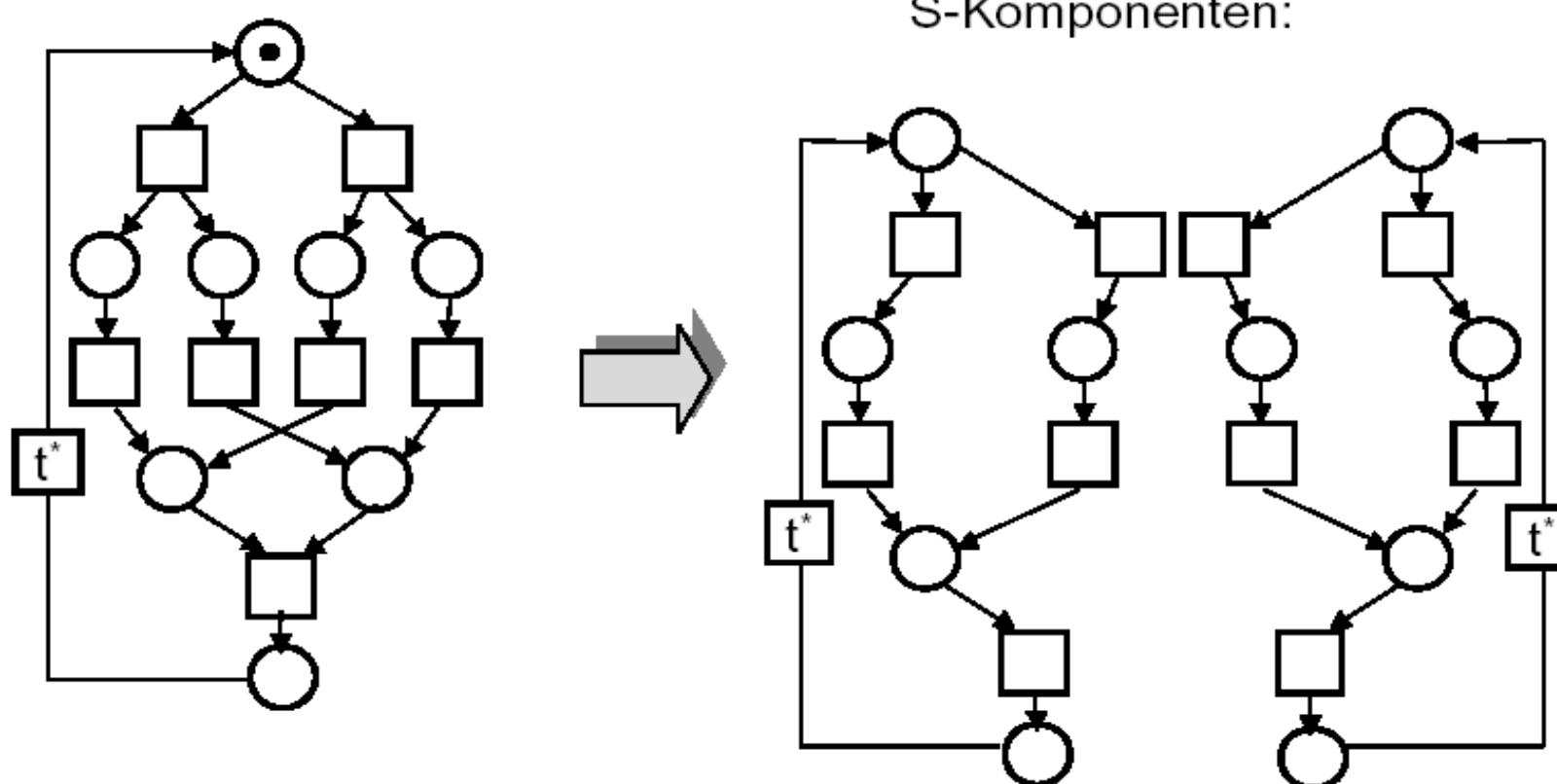
$$(q, t) \in F \Rightarrow (q, t) \in F' \text{ und } (t, q) \in F \Rightarrow (t, q) \in F'.$$

Besteht ein WF-Netz aus S-Komponenten, so wird es **s-coverable** genannt.

Bemerkung: Für ein s-coverable Petrinetz-Modell existiert eine Dekomposition in S-Komponenten, die Zustandsautomaten sind!

2. Struktur-Elemente und -Eigenschaften

Beispiel: S-Komponenten / S-coverable



lebendiges, beschränktes
free-choice Netz

→ S-coverable

Zusammenfassung:

- free-choice, well-structured, s-coverable sind sehr nützliche Struktureigenschaften zur Prozessanalyse,
- s-coverable sollte jede Workflow-Definition sein; diese Eigenschaft ist die Generalisierung der free-choice und well-structured-Eigenschaft,
- ein WF-Netz sollte mindestens entweder free-choice oder well-structured sein, um es effizient analysieren zu können (sonst oft Fehlerquelle).

3. Dynamische Elemente und Eigenschaften

Ziel: Allgemeine Aussagen über das Schaltverhalten des Netzes
→ Basis für formale Analyse dynamischer Systeme im Vorfeld der Implementierung

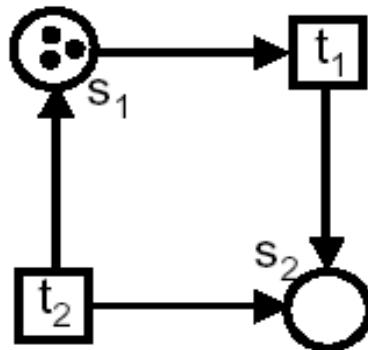
Wiederholung:

- Marken, Markierungen (Darstellung von Abläufen),
- Anfangsmarkierung, Folgemarkierung,
- Erreichbarkeit (erreichbare Markierung, Schaltfolge),
- Konflikt, Synchronisation,
- Trigger,
- unterscheidbare Marken,
- Zeiterweiterung.

3. Dynamische Elemente und Eigenschaften

Sei $N = (S, T, F)$ ein Petrinetz und m_0 eine Markierung von N ,
 m_0 wird **Anfangsmarkierung von N** genannt.

$$m_0 = (3, 0)$$

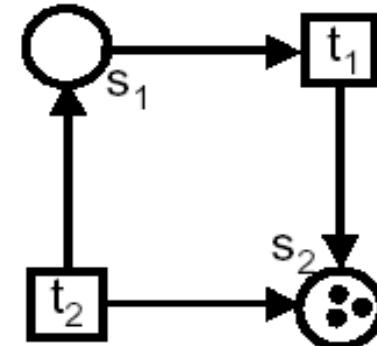
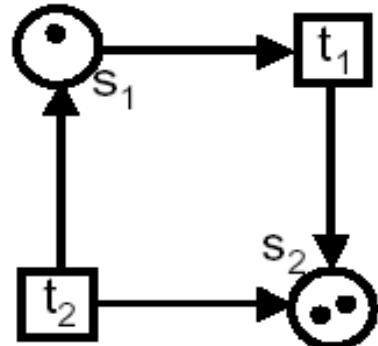
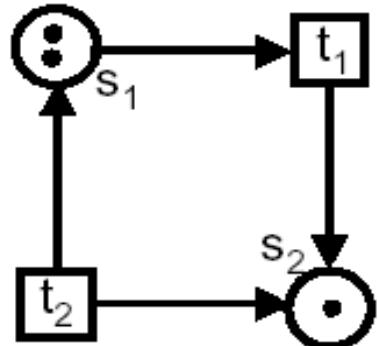


Ablaufzeitpunkte:

$$\tau_1 : m_1 = (2, 1)$$

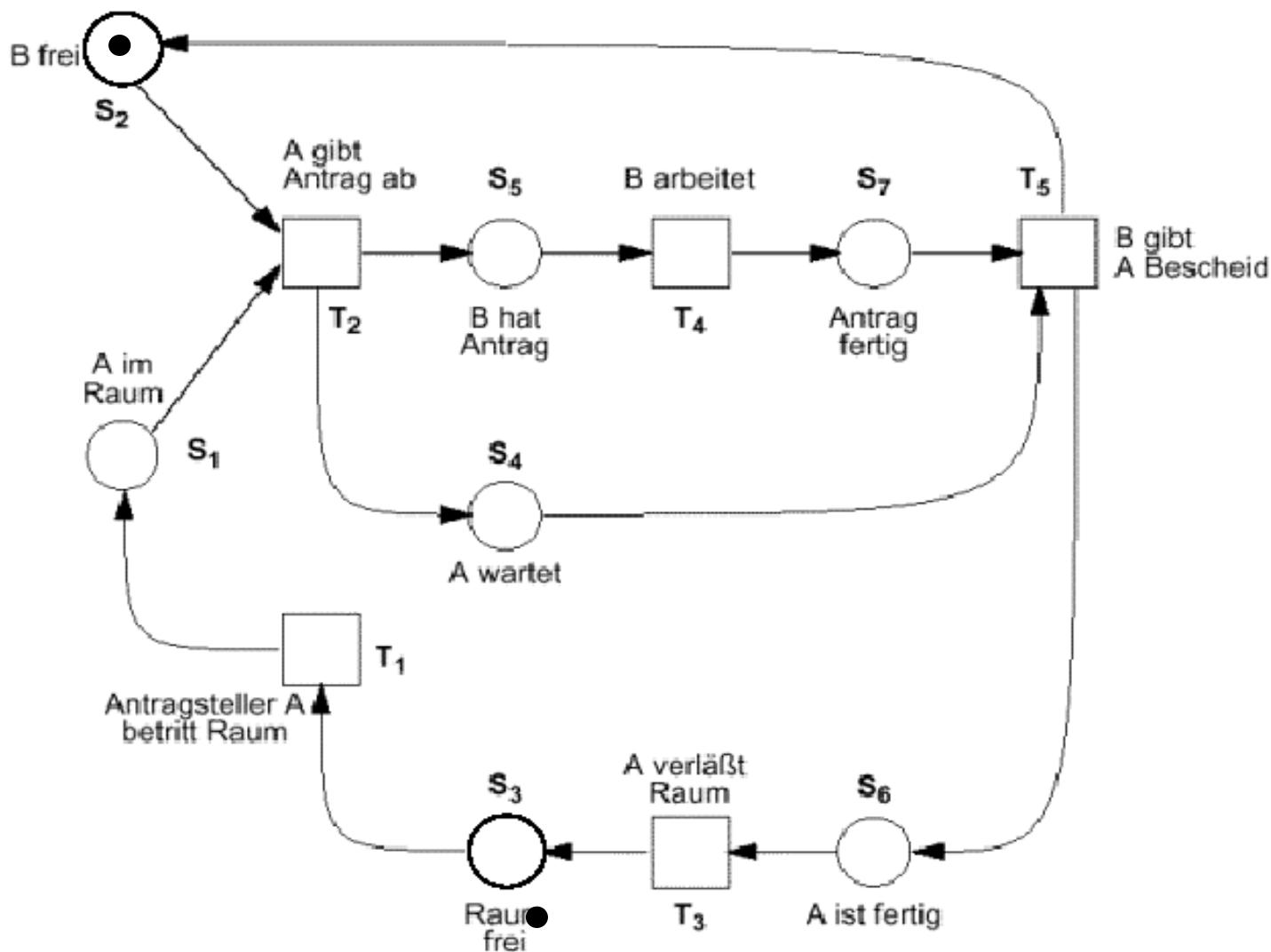
$$\tau_2 : m_2 = (1, 2)$$

$$\tau_3 : m_3 = (0, 3)$$

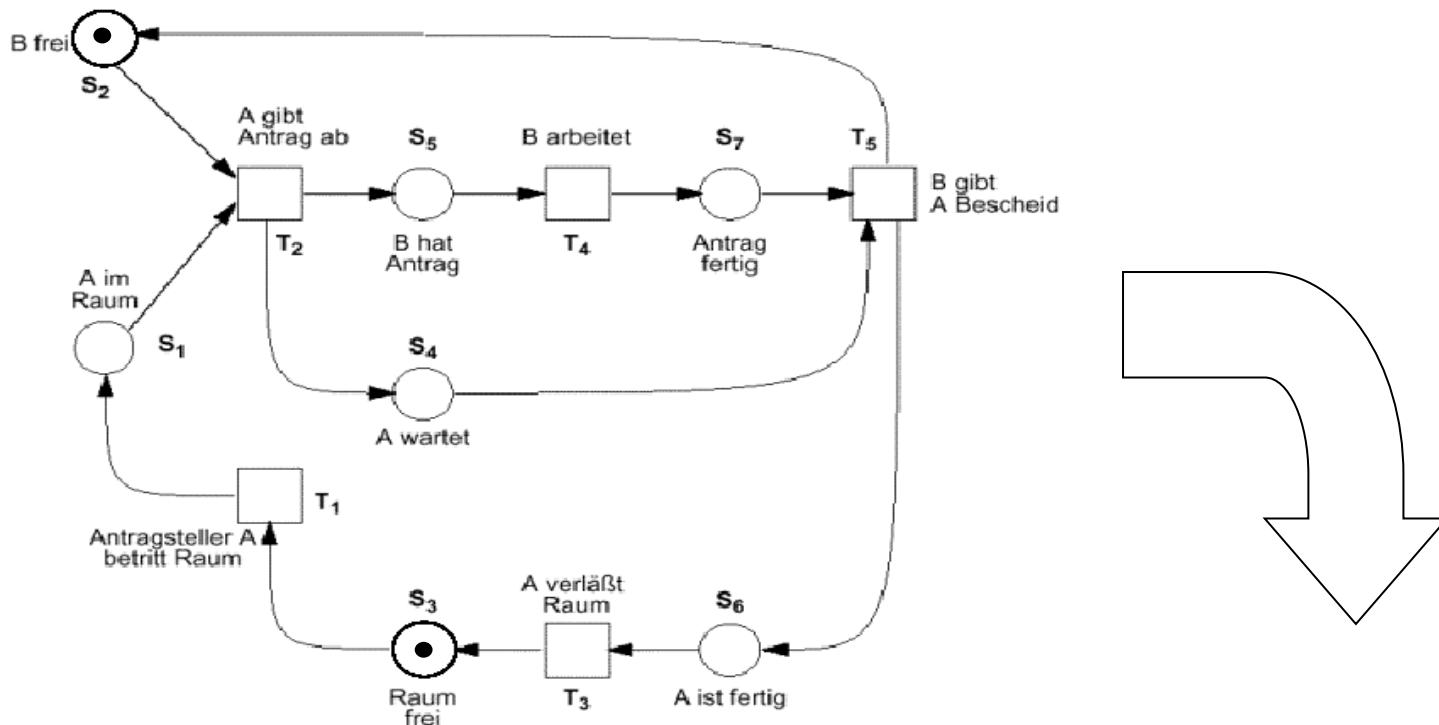


3. Dynamische Elemente und Eigenschaften

Beispiel: Schalterraum einer Behörde



3. Dynamische Elemente und Eigenschaften



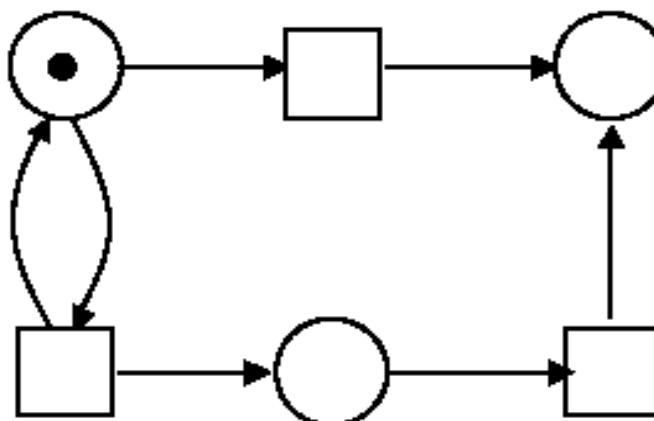
Zeitpunkt	Markierung der Stellen							Schaltende Transition				
	S ₁	S ₂	S ₃	S ₄	S ₅	S ₆	S ₇	T ₁	T ₂	T ₃	T ₄	T ₅
0		X	X					X				
1	X	X								X		
2				X	X							X
3					X					X		X
4			X								X	
5		X	X									

Beschränktheit

(N, m_0) heißt **beschränkt**, wenn eine Schranke b existiert,
so dass $m(s) \leq b$ für alle $s \in S$; für alle $m \in [m_0]$.

(N, m_0) heißt **1-beschränkt** oder **sicher**, wenn $b = 1$.

Frage: beschränkt oder unbeschränkt?



→ **unbeschränkt!**

Komplementbildung

(zur Erzwingung einer maximalen Anzahl von Marken in s)

Sei s eine Stelle;
eine neue Stelle s^* mit

$$s^* \cdot = \cdot s \setminus s \cdot$$

$$\cdot s^* = s \cdot \setminus \cdot s$$

$$m_0(s^*) = k(s) - m_0(s),$$

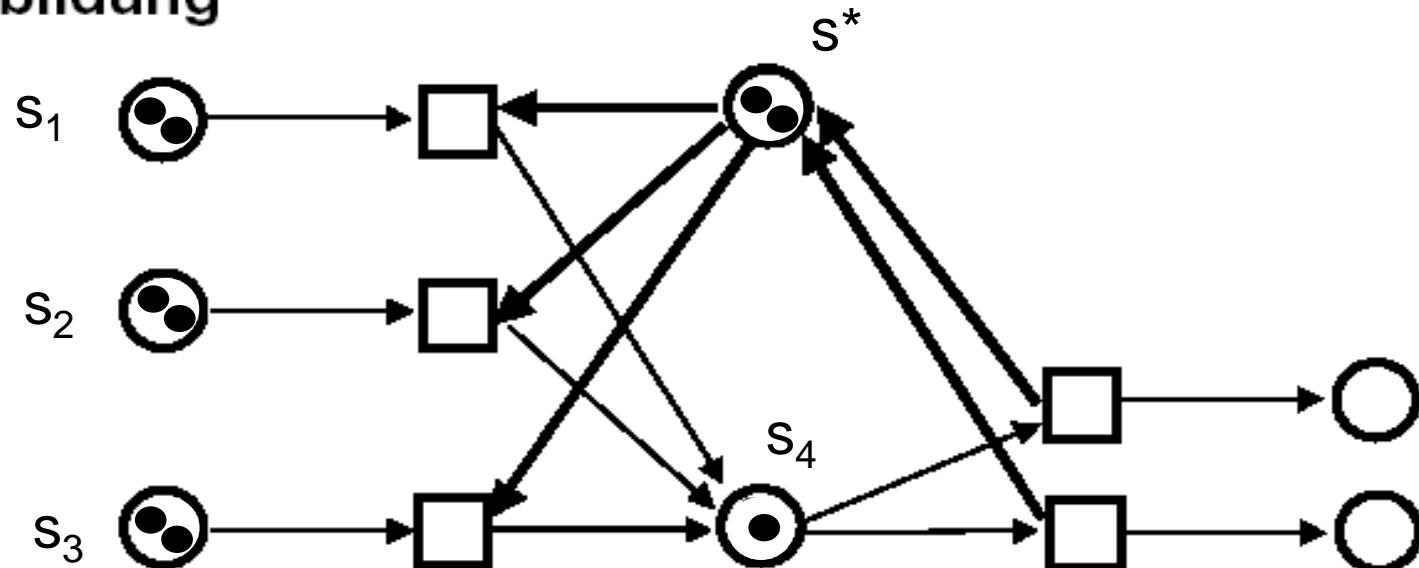
wird als das **Komplement** (die Komplementstelle) von s bezeichnet.

$k(s)$ ist die gewünschte Schranke für s

Es gilt: $\forall m \in [m_0 : m(s) = K(s) - m(s^*)$

d.h. $m(s) \leq K(s) \quad \forall m \in [m_0 :$

Komplementbildung



Ohne s^* könnten alle in den Stellen s_1 , s_2 und s_3 befindlichen Marken in s_4 erscheinen. Um dies zu umgehen, wird Komplementstelle s^* eingeführt, derart dass alle Input-Transitionen von s_4 Output-Transitionen von s^* werden und alle Output-Transitionen von s_4 Input-Transitionen von s^* .
Um eine Beschränkung mit der Schranke $b=3$ zu erreichen, muss $m_0(s^*)=b(s_4)-m_0(s_4)$ gelten. Somit muss die Stelle s^* Anfangs 2 Marken enthalten.

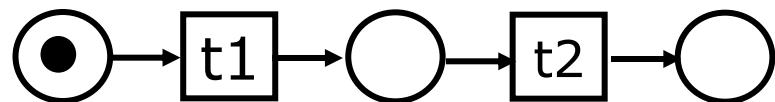
Tot, Verklemmungsfreiheit:

- Eine Transition t heißt **tot** unter einer Markierung m , wenn kein $m' \in [m]$ die Transition t aktiviert.
 - Eine Markierung heißt **tot**, wenn alle Transitionen tot sind.
 - Ein Petrinetz heißt **tot**, wenn seine Anfangsmarkierung m_0 tot ist.
 - Eine Markierung heißt **verklemmungsfrei**, wenn keine tote Markierung erreichbar ist.
- In einem System ist es sehr wichtig tote Markierungen zu erkennen und diese zu eliminieren, um einen einwandfreien Ablauf zu gewährleisten.

Lebendigkeit:

- Gibt es von jeder erreichbaren Markierung aus einen Weg zu jeder Transition?
- Eine Transition heißt **lebendig** unter einer Markierung m , wenn sie unter keiner Folgemarkierung $m' \in [m]$ tot ist.
- Eine Markierung m heißt **lebendig**, wenn alle Transitionen unter m lebendig sind.
- Ein Petrinetz heißt **lebendig**, wenn seine Anfangsmarkierung m_0 **lebendig** ist.

Tot ≠ nicht lebendig



Die Transitionen t_1 und t_2 sind beide nicht tot unter der Anfangsmarkierung $m_0=(1\ 0\ 0)$. Also ist auch m_0 nicht tot.

Allerdings sind beide Transitionen auch nicht lebendig, da sie unter der Folgemarkierung $m'=(0\ 0\ 1)$ tot sind. Somit ist auch die Anfangsmarkierung m_0 und damit auch das Petrinetz unter m_0 nicht lebendig!

Deadlock:

- Eine Markierung, nach deren Erreichen es keine Transition gibt, die noch schalten kann, heißt **Deadlock**.

Livelock:

- Eine Markierung, nach deren Erreichen keine Beendigung und auch kein Deadlock mehr möglich ist, heißt **Livelock**.

Reversibel:

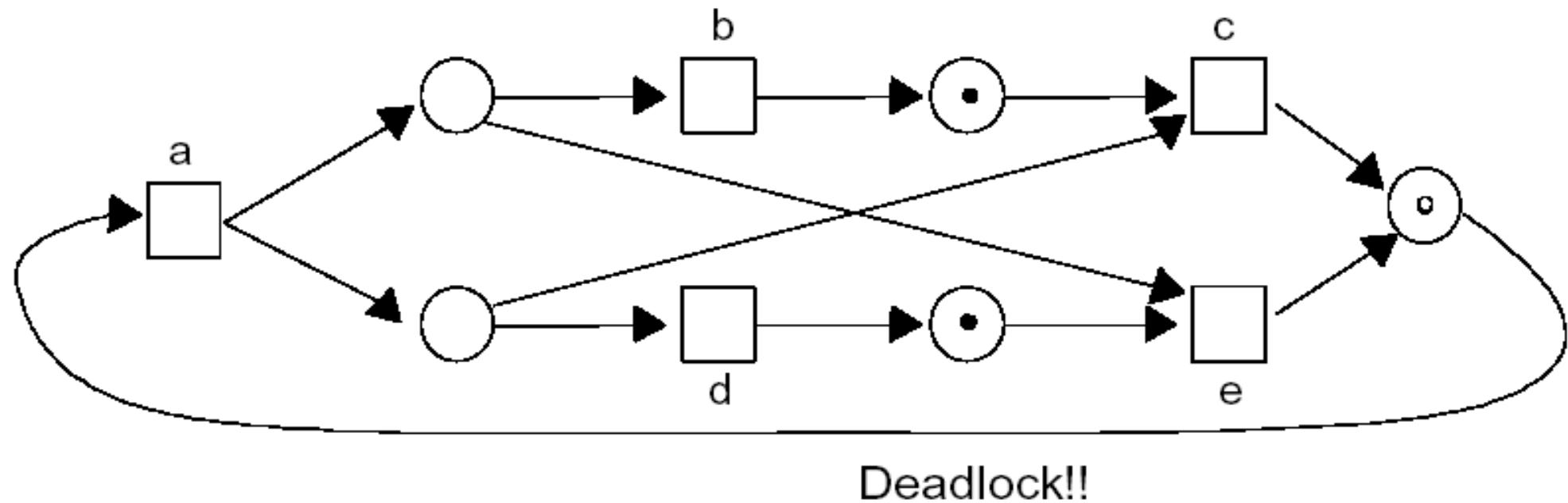
- Ein markiertes Petrinetz mit Anfangsmarkierung m_0 heißt **reversibel**, wenn m_0 von jeder erreichbaren Markierung aus erreichbar ist.

Terminierung:

- Ein markiertes Petrinetz **terminiert**, wenn die Menge der Schaltfolgen endlich ist.

3. Dynamische Elemente und Eigenschaften

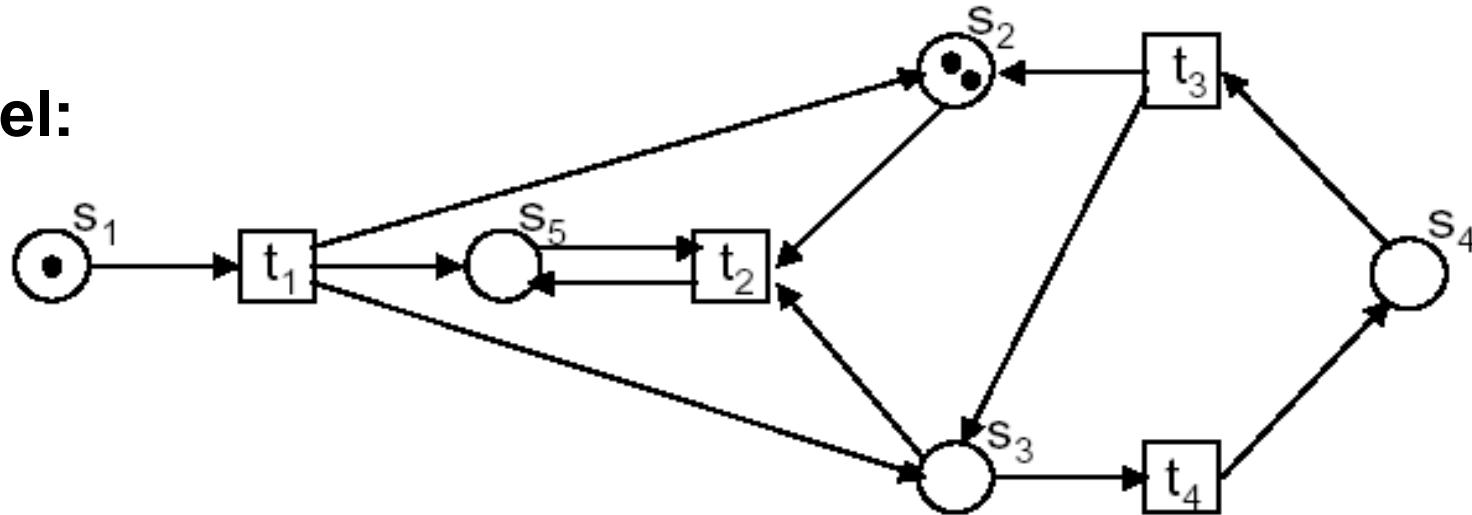
Beispiel:



beliebig lange Schaltfolgen (a b c) möglich, aber nicht verklemmungsfrei (a b d)

3. Dynamische Elemente und Eigenschaften

Beispiel:



1. Tote Transitionen unter m_0 : keine,
2. Deadlock, durch Schalten von t_1 und t_2 ,
3. Lebendigkeit, nicht gegeben durch die Anwesenheit eines Deadlocks,

Eigenschaften in Sätzen:

- Jedes lebendige markierte Petrinetz mit mindestens einer Transition ist verklemmungsfrei.
- Ein markiertes Petrinetz ist genau dann lebendig, wenn unter keiner erreichbaren Markierung eine tote Transition existiert.
- Ein markiertes Petrinetz ist genau dann beschränkt, wenn die Menge der erreichbaren Markierungen endlich ist.

3. Dynamische Elemente und Eigenschaften

Entscheidend für einen Workflow-Prozess ist die Eigenschaft **Soundness**
(minimale Anforderungen an einen Workflow)

- Beendigungsmöglichkeit:
Es sollte immer möglich sein einen Fall zu beenden. Dies garantiert, dass keine Deadlocks vorhanden sind.
- Richtigkeit der Beendigung:
Das Prozessende sollte eindeutig sein, d.h. nach Beendigung eines Falles sollten keine Aufgaben für diesen Fall mehr auszuführen sein.
- Aufgabenerfordernis:
Jede Aufgabe sollte die Möglichkeit haben ausgeführt zu werden, sollte also für den Prozess erforderlich sein.

Schlussfolgerung für das Petrinetz:

- Für jeden Fall (Case) terminiert das entsprechende WF-Netz irgendwann.
- In diesem Zustand befindet sich genau eine Marke in der Stelle o , alle anderen Stellen sind leer.
(Bezeichnung: Markierung $o=(0\ 0 \dots\ 1)$; analog: Markierung i für den Anfangszustand)
- Es sollten keine toten Transitionen existieren, d.h. jede Aufgabe (Task) sollte beim Durchlaufen eines geeigneten Pfades im WF-Netz ausführbar sein.

3. Dynamische Elemente und Eigenschaften

Soundness (Stabilität, Zuverlässigkeit)

Eine durch ein WF-Netz $N = (P, T, F)$ modellierte Prozessdefinition wird als **sound** bezeichnet genau dann, wenn gilt:

- für jede von i aus erreichbare Markierung M existiert eine Schaltfolge von M nach o , d.h.
 $\forall M: i \xrightarrow{*} M \Rightarrow M \xrightarrow{*} o$
- die Markierung o ist die einzige von i erreichbare Markierung mit mindestens einer Marke in o , d.h.
 $\forall M: (i \xrightarrow{*} M \wedge M \geq o) \Rightarrow M = o$
- es existieren keine toten Transitionen, d.h.
 $\forall t \in T \exists M, M': i \xrightarrow{*} M \xrightarrow{t} M'$

Soundness:

- Ein WF-Netz ist sound genau dann, wenn es lebendig und beschränkt ist.
- Soundness lässt sich mit Standard-Methoden für Petrinetze nachweisen.
- Für Free-Choice-Netze kann Soundness in polynomieller Zeit gezeigt werden.

Soundness ist eine Minimalanforderung.

Für komplexe WF-Netze ist die Entscheidbarkeit von Soundness ein exponentiell hartes Problem.

Die Definition von Soundness liefert keine Anhaltspunkte für Verbesserungen der Prozessdefinition.

Invarianten:

- Als Invariante eines Systems bezeichnet man eine solche Eigenschaft, die bei der Arbeit des Systems unabhängig vom konkreten Ablauf erhalten bleibt.

Stellen-Invariante (kurz S-Invariante):

- formuliert Bedingung über eine gewisse Konstantheit der Anzahl der Token eines Netzes N.

Transitions-Invariante (kurz T-Invariante):

- ist Folge von Transitionen die von einer Markierung m aus geschaltet werden kann, um wieder die gleiche Markierung m zu erreichen.

3. Dynamische Elemente und Eigenschaften

- Eine S-Invariante Y weist jeder Stelle eine **Gewichtung** zu, derart, dass die **gewichtete Summe** der Marken für alle Markierungsübergänge **konstant** bleibt.

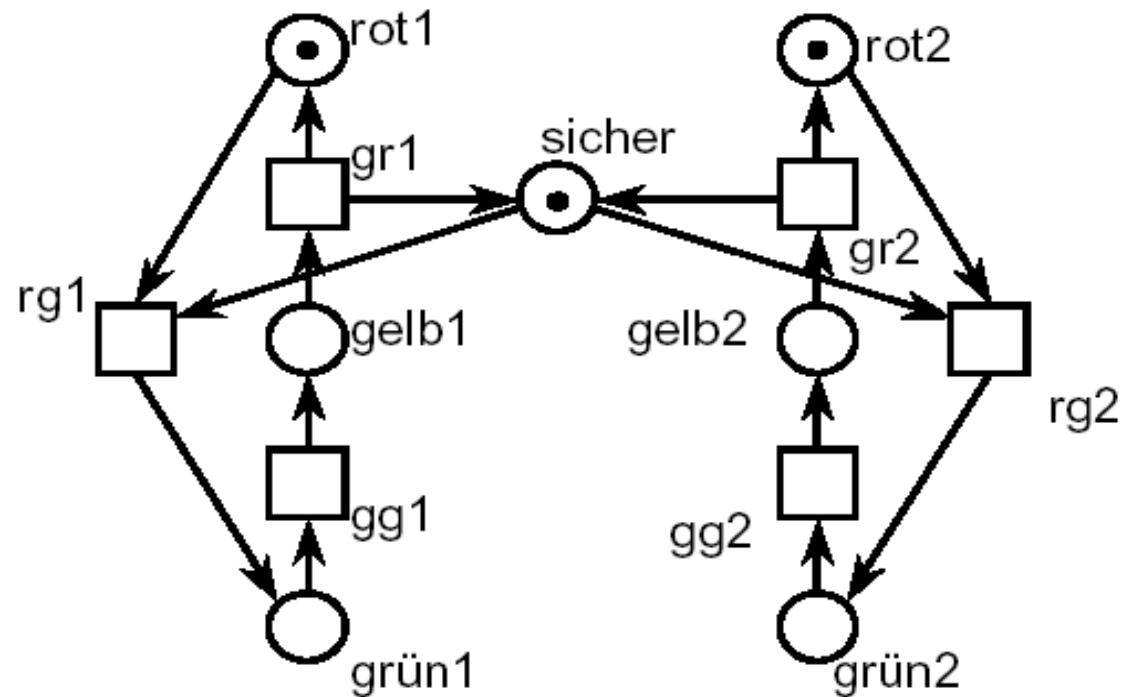
Sei $N=(S,T,F,m_0)$ ein Petrinetz mit der Anfangsmarkierung m_0 und $|S|=n$ ($s \in S$). Dann heißt eine Gewichtung $Y=(y_1(s_1), y_2(s_2), \dots, y_n(s_n))$ mit den Gewichten y_i der Stellen s_i S-Invariante, wenn:

$$\sum_{i=1}^n y_i(m(s_i)) = \text{const.}, \forall m \in [m_0 >$$

- D.h. bei einer S-Invarianten Y ist die mit Y gewichtete Tokenzahl invariant gegenüber dem Schalten.
- Für jede erreichbare Markierung entspricht die gewichtete Summe der Marken der gewichteten Summe der Marken der Anfangsmarkierung (eine Konstante).

3. Dynamische Elemente und Eigenschaften

Stellen-Invarianten: Beispiel



$$\text{rot1} + \text{gelb1} + \text{grün1} = 1 \quad (1, 1, 1, 0, 0, 0, 0)$$

$$\text{rot2} + \text{gelb2} + \text{grün2} = 1 \quad (0, 0, 0, 1, 1, 1, 0)$$

$$\text{sicher} + \text{grün1} + \text{grün2} + \text{gelb1} + \text{gelb2} = 1 \quad (0, 1, 1, 0, 1, 1, 1)$$

$$\text{rot1} + \text{rot2} - \text{sicher} = 1 \quad (1, 0, 0, 1, 0, 0, -1)$$

Transitions-Invarianten:

- Eine T-Invariante $X = (x(t_1), x(t_2), x(t_3), \dots, x(t_n))$ gibt an, wie oft welche Transitionen von einer Markierung m aus geschaltet werden müssen, um wieder die gleiche Markierung m zu erreichen.

d.h.
- Eine T-Invariante ist eine Schaltfolge $\tau = t_1 t_2 \dots t_n$ von m nach m' , so dass gilt :
 $m(s_i) = m'(s_i)$, für alle i .

Geschäftsprozess-Management

Prof. Dr.-Ing. Andreas Ittner

Email: ittner@hs-mittweida.de

WWW: www.hs-mittweida.de/~ittner

Tel.: +49(0)3727-58-1288

Mob.: +49(0)177-5555-347

Fax.: +49(0)180-35518-22467

Gliederung

- Motivation
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs, BPMN, ...
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und Performance-Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards

1. Fragestellungen und klassische Fehler,
2. Struktur-Elemente und -Eigenschaften (Zusammenhängend, Free Choice, Workflow-Netz, well-handled, -structured, S-Komponenten / s-coverable),
3. Dynamische Elemente und Eigenschaften (Anfangsmarkierung, Beschränktheit, Komplementbildung, tot, lebendig, verklemmungsfrei, Soundness, Invarianten (S-, T-Invarianten)),
4. Erreichbarkeitsanalyse,
5. Linear Algebraische Darstellung,
6. Zusammenhänge bei der Prozess-Analyse.

4. Erreichbarkeitsanalyse

Analyse der

- erreichbaren Zustände,
- des möglichen Verhaltens.

→ aufwendig bis unmöglich (wenn zu komplex).

Prozessmodelle und Eigenschaften können nur für einfache Beispiele „von Hand“ überprüft werden.

Vorteil des Petrinetz-basierten Ansatzes

- erfüllt Anforderungen an Modellierungssprache,
- viele Methoden zur Analyse,
- viele Werkzeuge zur „automatischen“ Verifikation.

4. Erreichbarkeitsanalyse

Der Graph aus den Knoten $[m_0]$ und mit den Transitionen beschrifteten Kanten $\{(m, t, m') \mid m \rightarrow m'\}$ heißt **Markierungsgraph** von (N, m_0) .

Alle möglichen Markierungen des Netzes werden als Knoten dargestellt, die Übergänge zu Markierungen als Kanten.

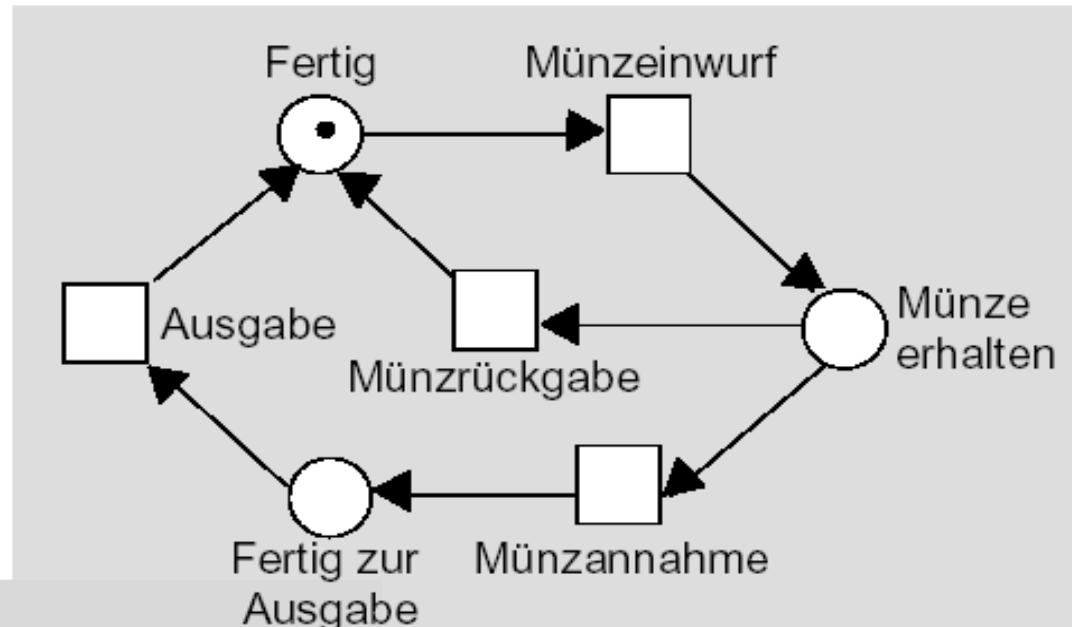
Die Anfangsmarkierung wird im Markierungsgraph besonders dargestellt: $\rightarrow \bullet$.

- Darstellung möglicher erreichbarer Markierungen / Zustände,
- Erkennen von Deadlocks

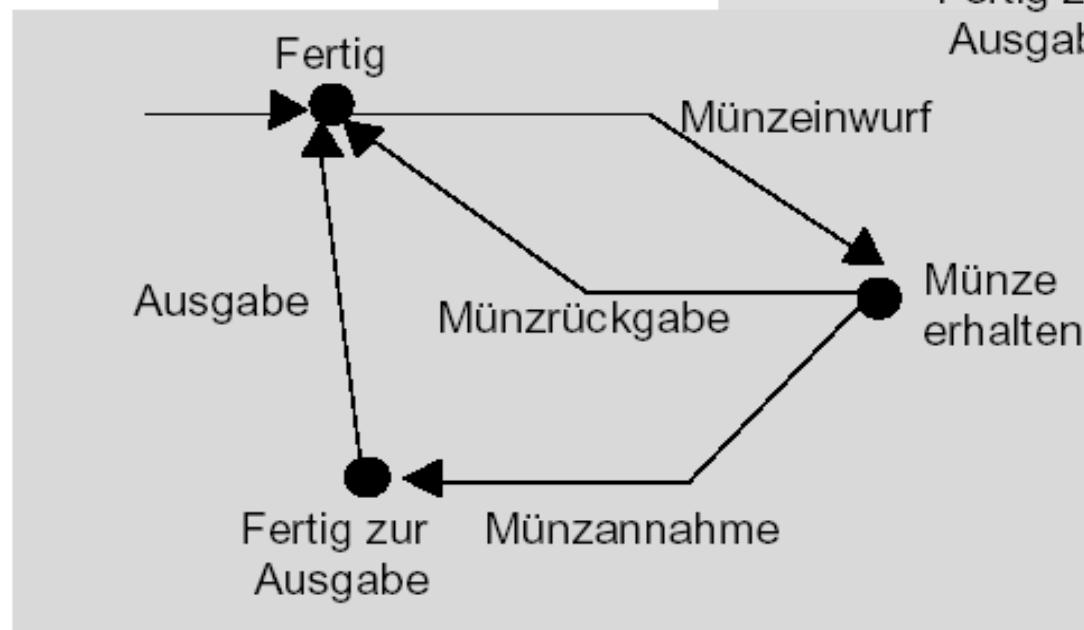
Verwendetes Synonym: Erreichbarkeitsgraph

4. Erreichbarkeitsanalyse

Beispiel: Automat (I)



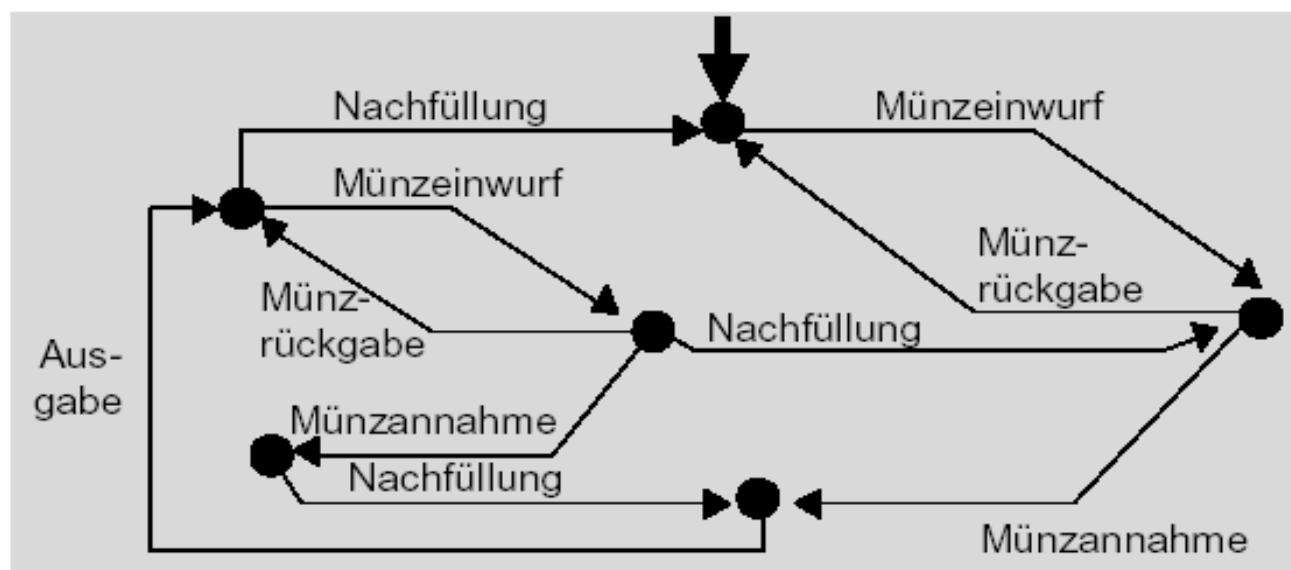
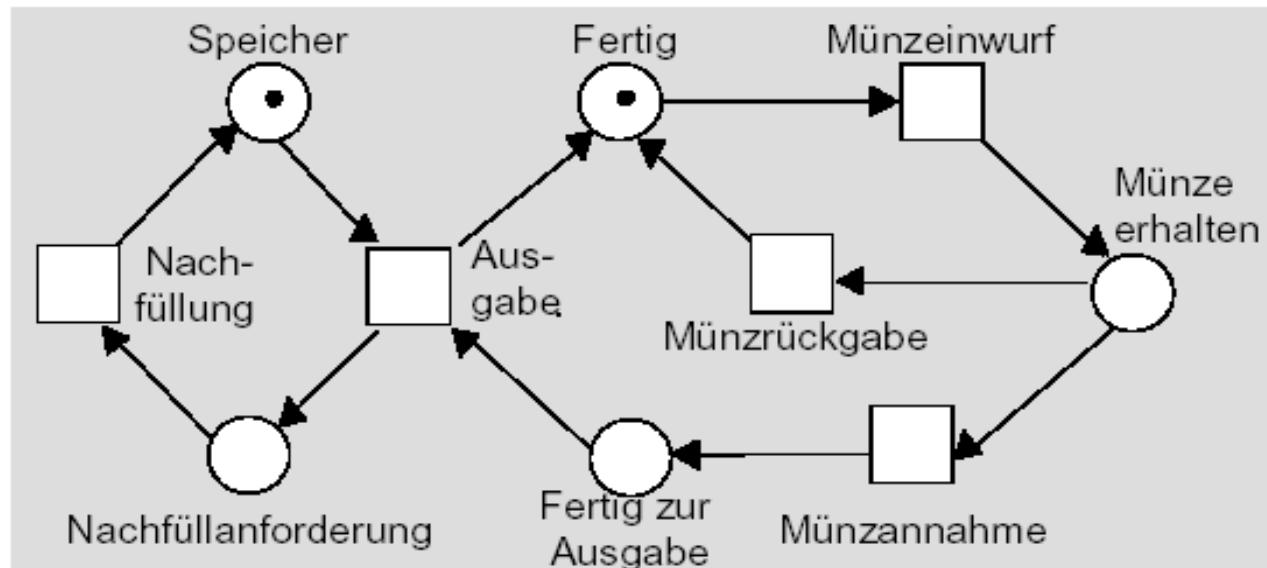
ein Automat durch ein
markiertes Petrinetz modelliert



das Verhalten des Automaten
als Markierungsgraph

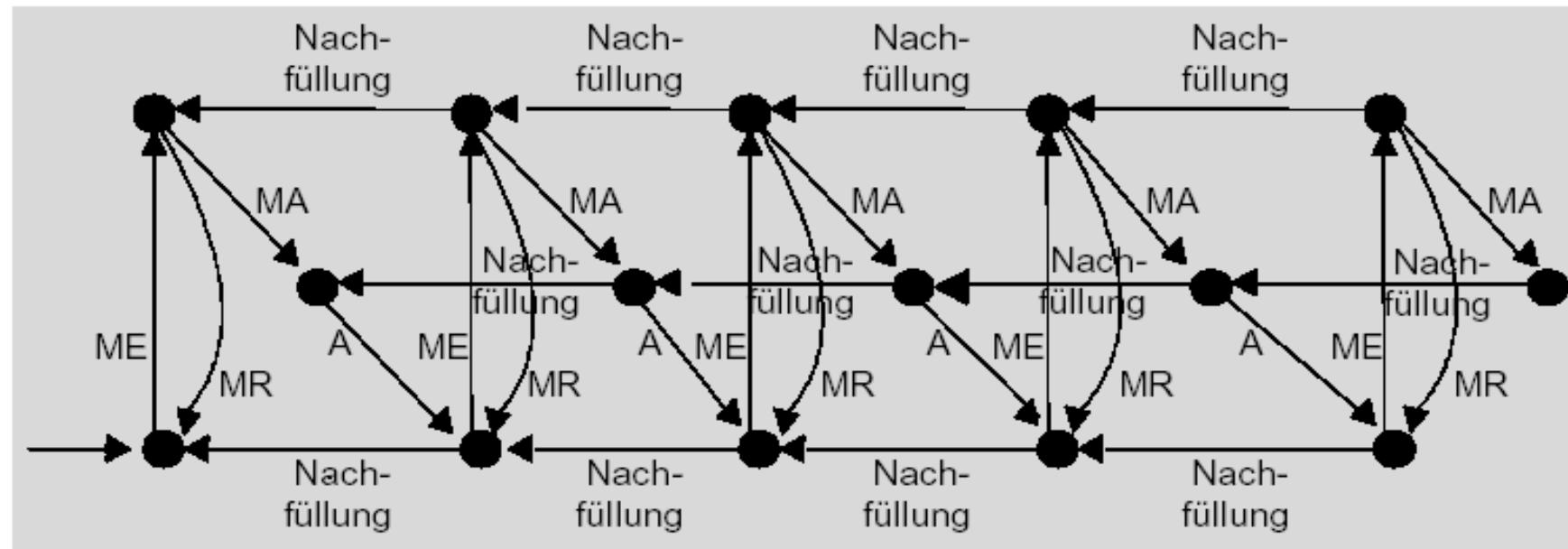
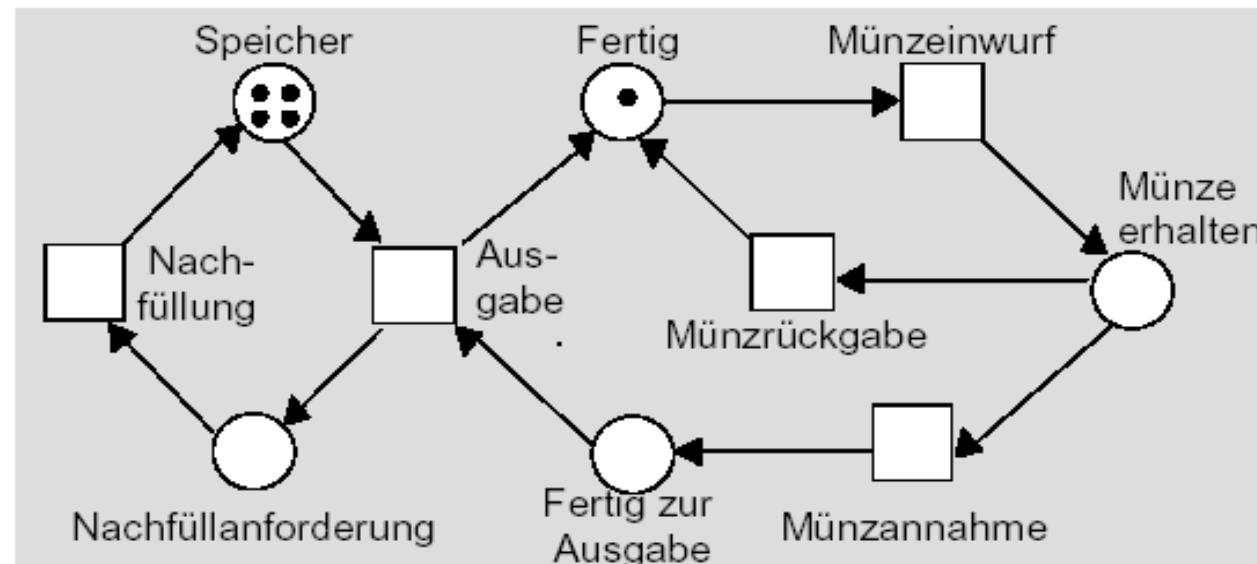
4. Erreichbarkeitsanalyse

Beispiel:
Automat (II)
mit Kapazität 1



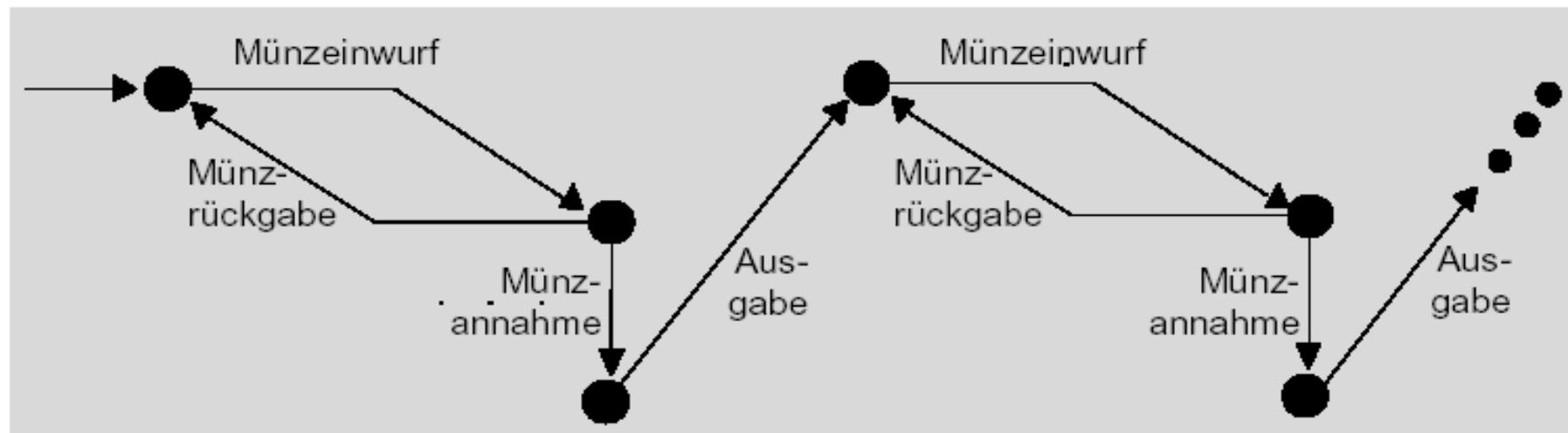
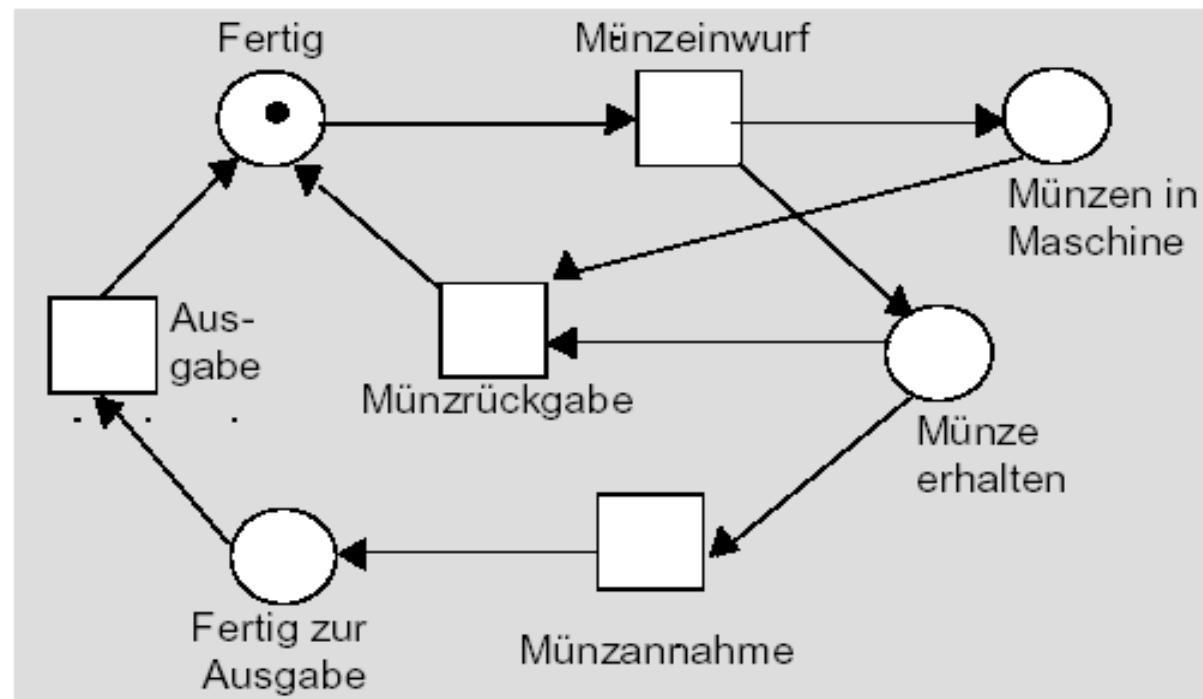
4. Erreichbarkeitsanalyse

Beispiel:
Automat (III)
mit Kapazität 4

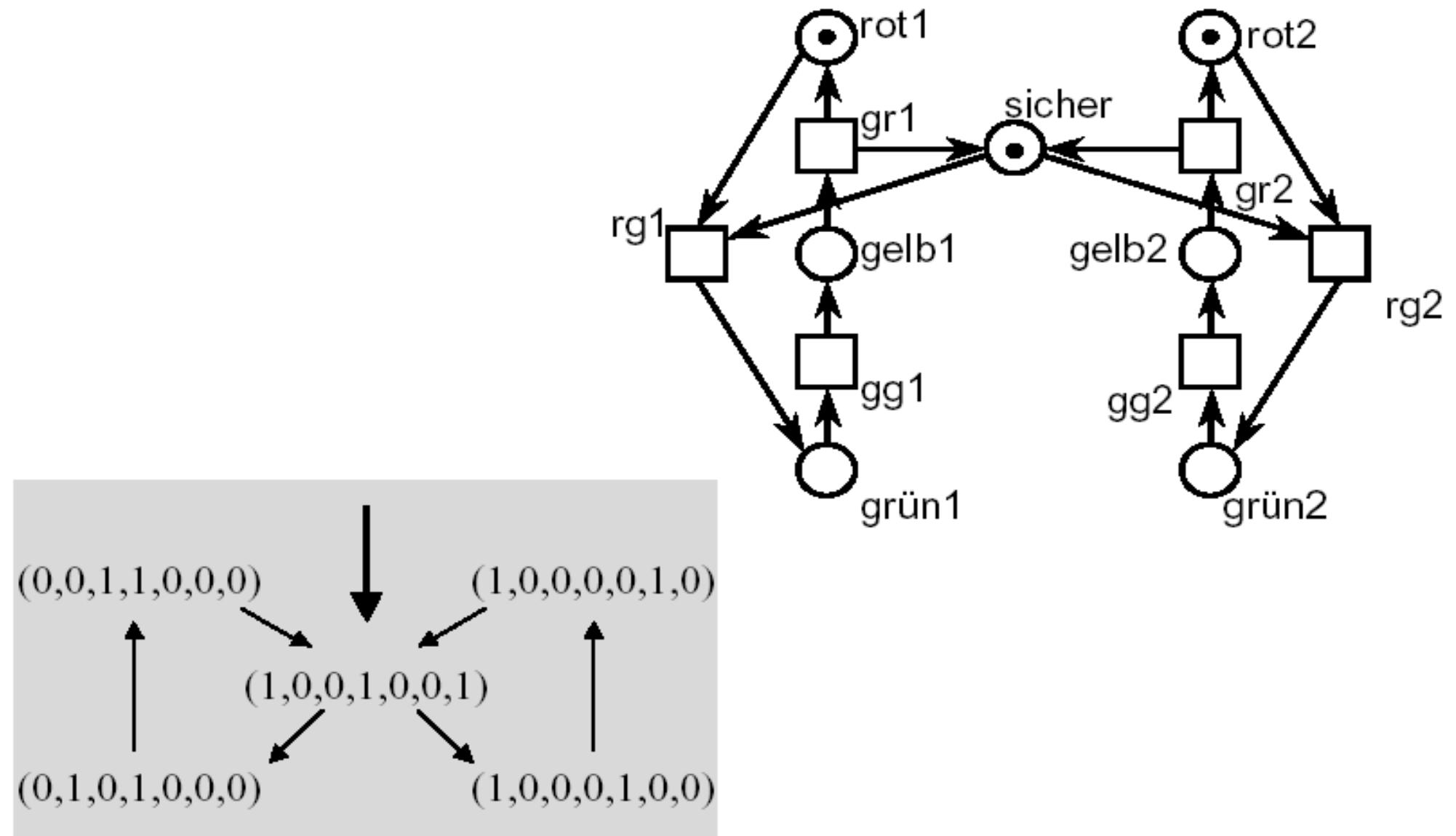


4. Erreichbarkeitsanalyse

Beispiel:
Automat (IV)
mit Zähler
(unbeschränkt)



4. Erreichbarkeitsanalyse



Tupel=(rot1, grün1, gelb1, rot2, grün2, gelb2, sicher)

4. Erreichbarkeitsanalyse

Eigenschaften des Markierungsgraphen:

1. Ein endlicher Markierungsgraph bedeutet, dass das Netz beschränkt ist. D.h., dass kein Überlauf von Marken in Stellen vorkommt.
2. Besitzt jeder Knoten des Markierungsgraphen einen Nachfolger, so ist das entsprechende Petrinetz verklemmungsfrei, also ist im System kein Stillstand zu erwarten.
3. Ein stark zusammenhängender Markierungsgraph bedeutet immer eine Reversibilität des Petrinetzes, also die Rücksetzbarkeit des Systems in den Anfangszustand.
4. Ist eine Markierung m Knoten des Markierungsgraphen, so heißt das für das Petrinetz, dass die Markierung m erreichbar ist.
Der entsprechende Zustand des Systems, den diese Markierung modelliert, ist erreichbar.

4. Erreichbarkeitsanalyse

- Problem Markierungsgraph: Anzahl der Zustände kann explodieren,
- Für beschränkte Netze kann die Zahl erreichbarer Markierungen exponentiell in der Größe des Netzes wachsen (Anzahl der Knoten),
- Eine Analyse des Verhaltens mittels Konstruktion des Markierungsgraphen ist deshalb praktisch unmöglich.
- → deshalb Einführung eines **Überdeckungsgraphen**

4. Erreichbarkeitsanalyse

Überdeckungsgraph

- kann für jedes - auch unbeschränktes - Petri-Netz konstruiert werden,
- ist die endliche Variante eines Markierungsgraphen,
- ist aber nicht eindeutig.

Grundidee:

- aussortieren streng monoton steigender Folgen von Markierungen aus der Erreichbarkeitsmenge,
- abbilden einer Menge von unendlich vielen Zuständen auf einen (einzigsten) „Ersatzzustand“.

! Information über die erreichbaren Zustände gehen verloren

4. Erreichbarkeitsanalyse

- x, z : Knoten des Markierungsgraphen,
- $m[x], m[z]$: zugehörige Markierungen.

Eine **Markierung $m[z]$** ist **größer** als eine **Markierung $m[x]$** , wenn alle Komponenten der Markierung von z größer oder gleich groß sind und eine Komponente echt größer ist, d.h. falls $m_i[z] \geq m_i[x]$ für alle i existiert ein j : $m_j[z] > m_j[x]$.

Alle Transitionen, die unter einer Markierung aktiviert sind, sind auch unter jeder größeren Markierung aktiviert!

Eine Markierung $m[z]$ eines Knotens z aus dem Markierungsgraphen **überdeckt** die Markierung $m[x]$ des Knotens x aus dem Markierungsgraphen, wenn gilt: Markierung $m[z] >$ Markierung $m[x]$.

4. Erreichbarkeitsanalyse

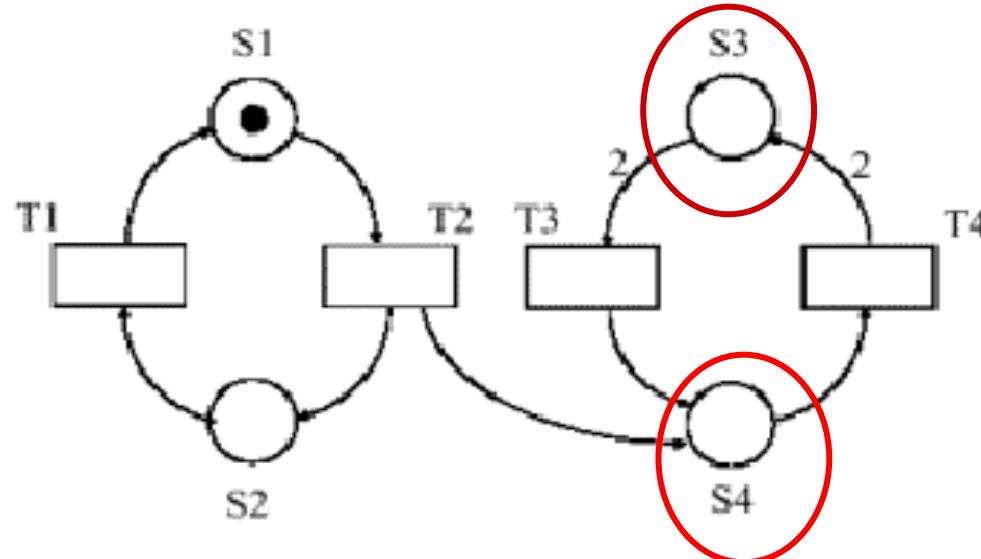
Überdeckungsgraph:

Enthält nur eine Teilmenge der Markierungen (Knoten) aus dem Markierungsgraphen. Zusätzlich aber gibt es spezielle Markierungen, die ein oder mehrere „w“ enthalten, wenn eine oder mehrere Stellen im Petrinetz existieren, die unbeschränkt sind.

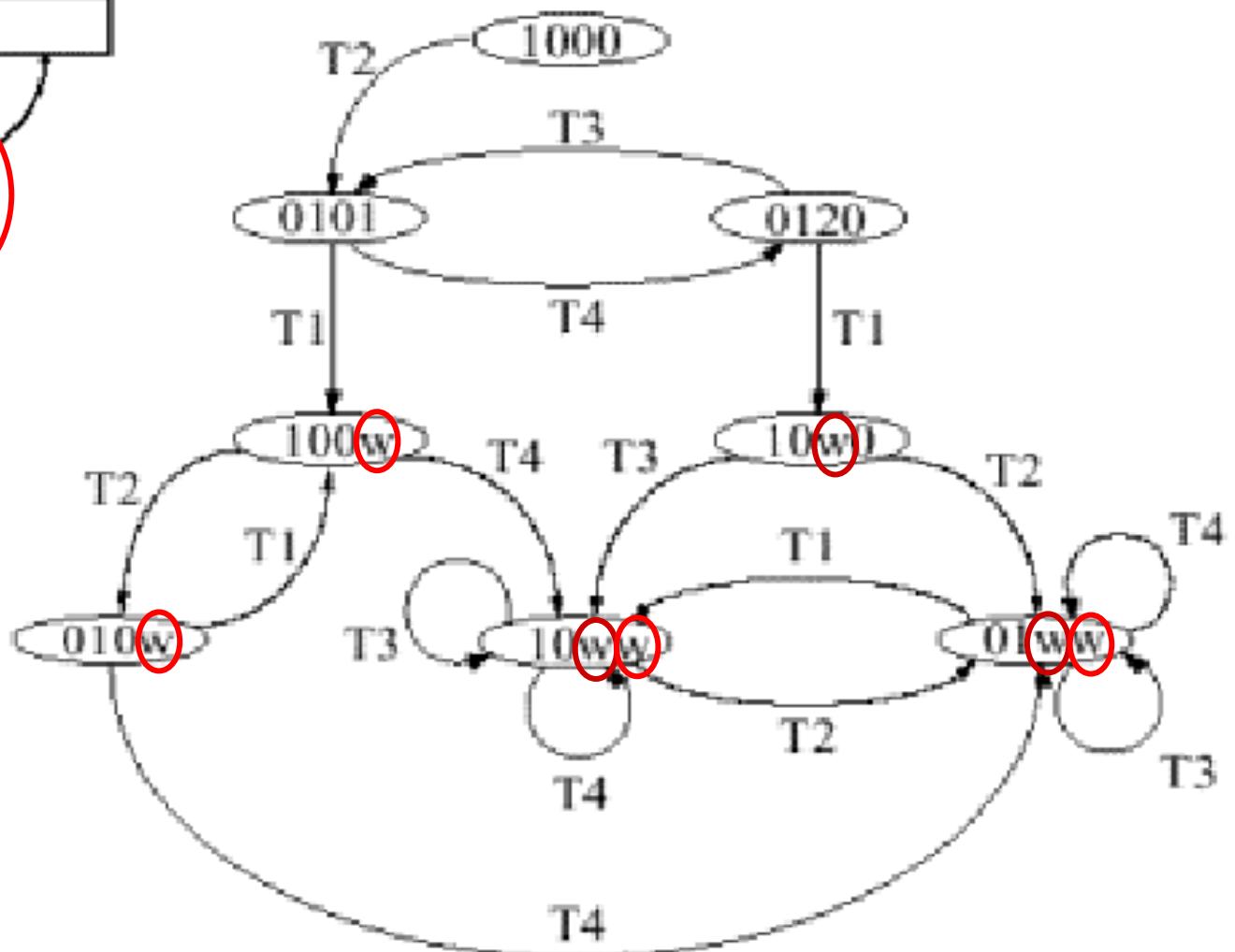
Mit einem „w“ gekennzeichnete Stellen sind Stellen, die unendlich viele Marken enthalten können.

Wir bezeichnen eine Markierung, die ein oder mehrere w enthalten kann als **w-Markierung**.

4. Erreichbarkeitsanalyse

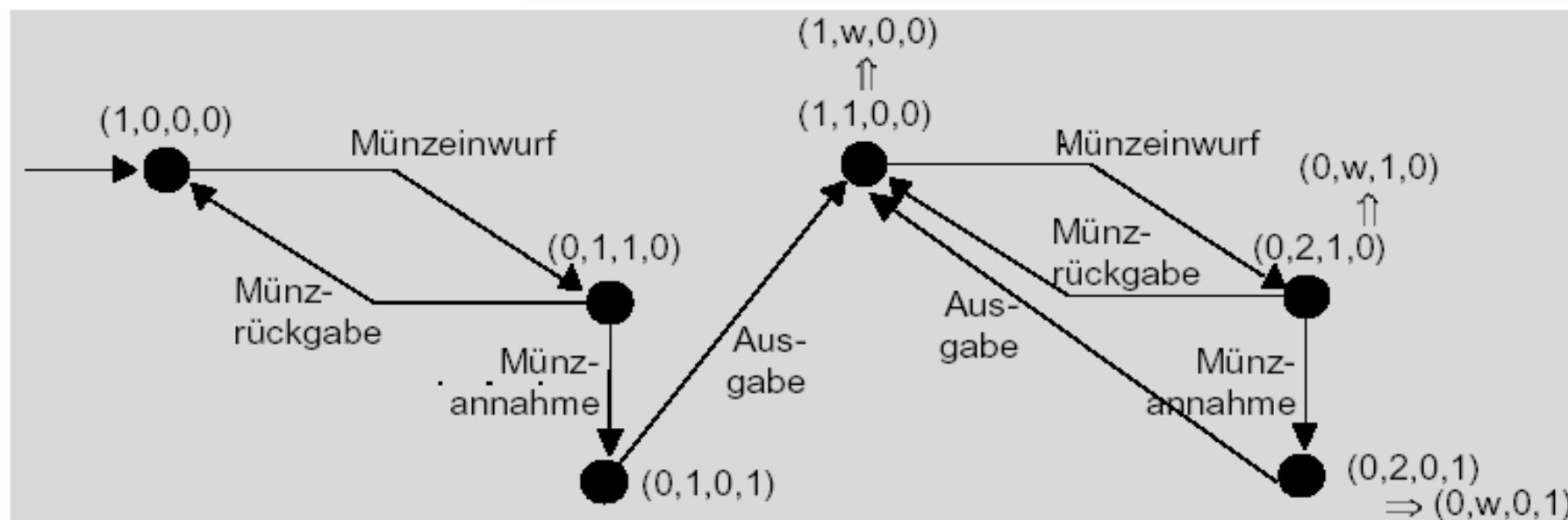
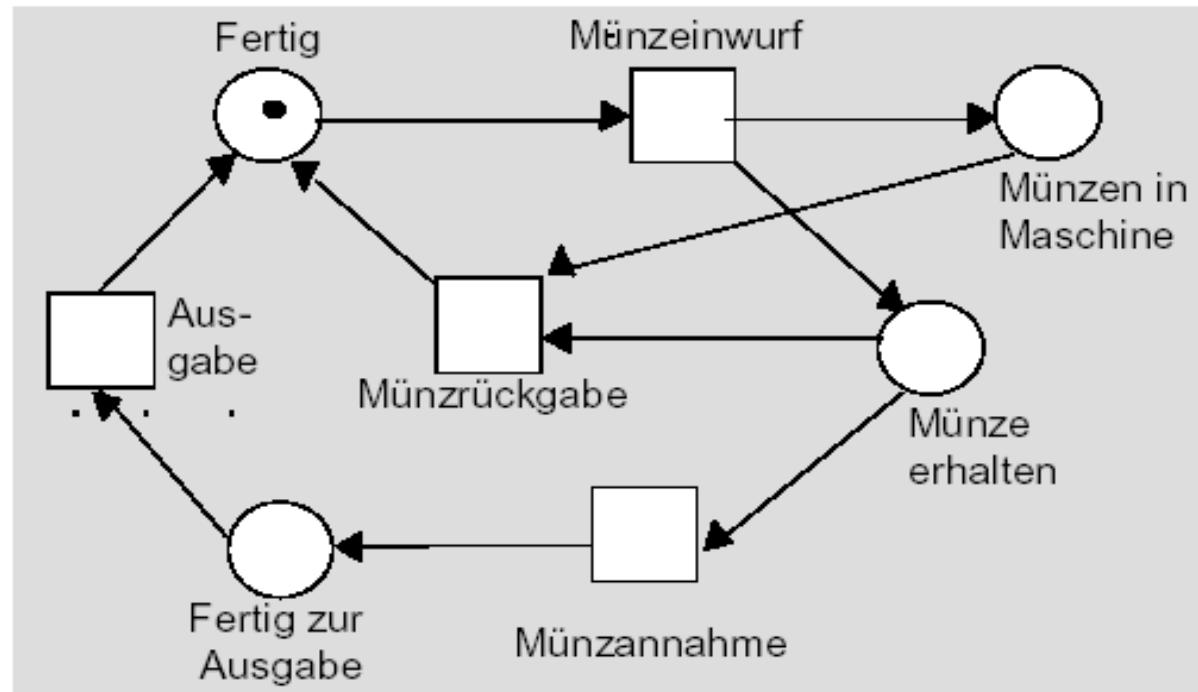


Beispiel: Überdeckungsgraph



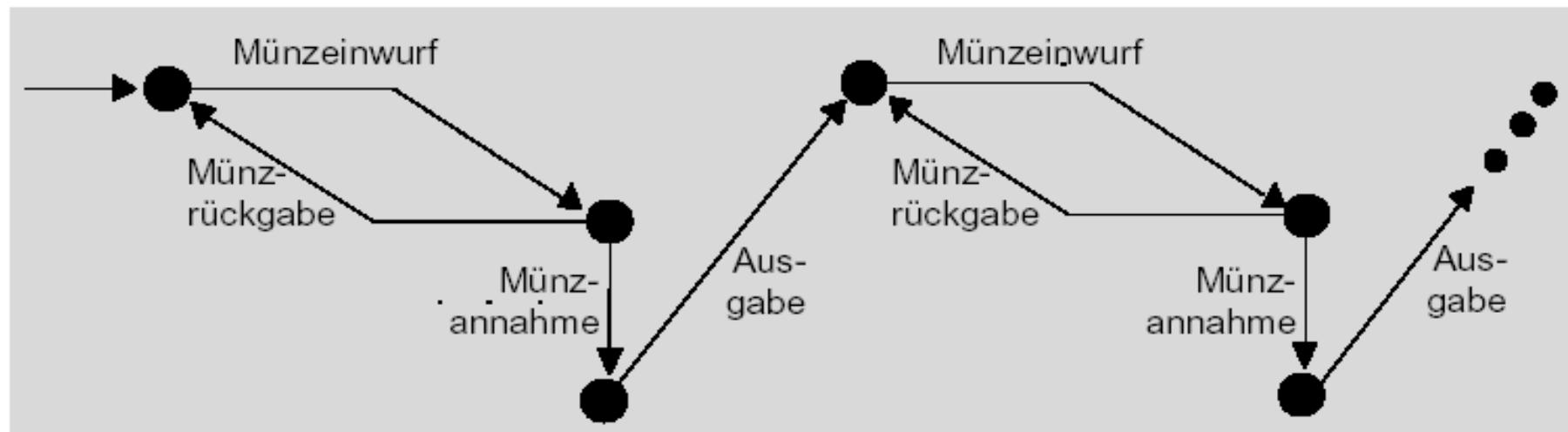
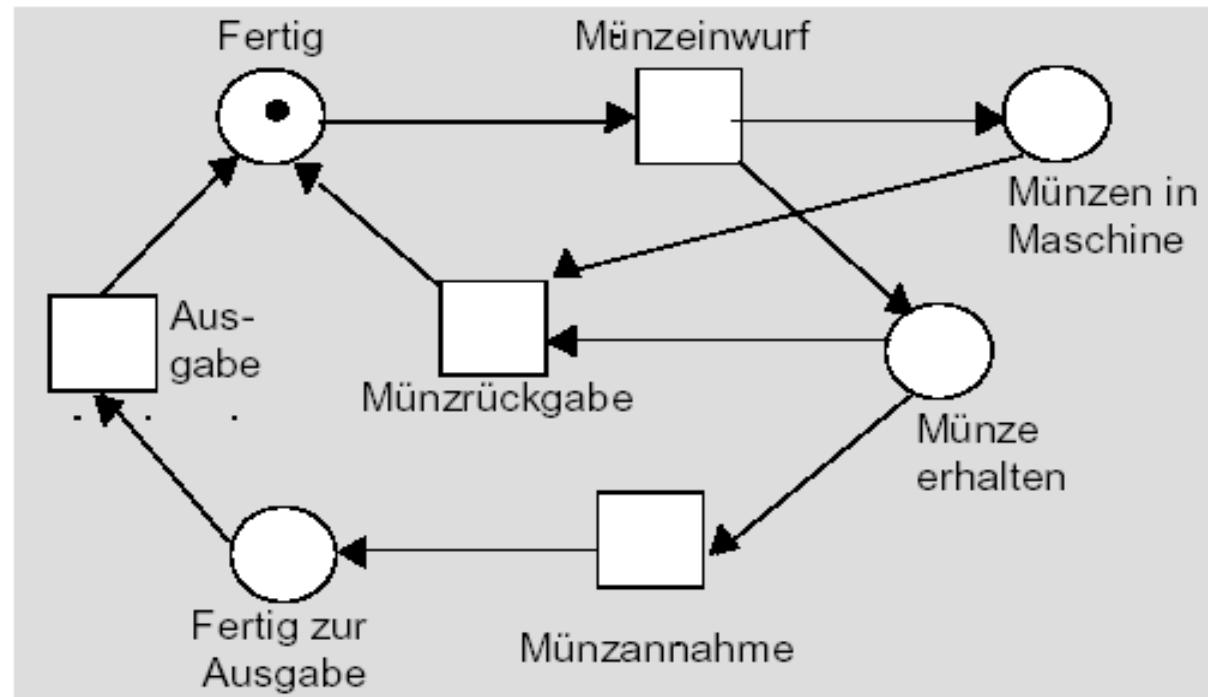
4. Erreichbarkeitsanalyse

Beispiel:
Überdeckungsgraph
Automat (IV)
mit Zähler
(unbeschränkt)



4. Erreichbarkeitsanalyse (Vergleich mit Markierungsgraph)

Beispiel:
Automat (IV)
mit Zähler
(unbeschränkt)



4. Erreichbarkeitsanalyse

Erzeugung des Überdeckungsgraphen zu (N,m0):

Initialisierung:

Anfangsknoten := Anfangsmarkierung m_0

Algorithmus:

Ermitteln der Folgemarkierung(en)

- **Fall 1:** es existiert schon ein Knoten mit derselben Beschriftung
→ Kante zu diesem Knoten hinzufügen
- **Fall 2:** es existiert schon ein Knoten $m[x]$, so dass die Folgemarkierung $m[z]$ größer ist
 - für die i mit $m_i[z] = m_i[x]$: Folgemarkierung erhält dieselbe Beschriftung wie Knoten $m[x]$
 - für die i mit $m_i[z] > m_i[x]$: Beschriftung := w .

4. Erreichbarkeitsanalyse

Erzeugung des Überdeckungsgraphen zu (N, m_0) (Fortsetzung):

- **Fall 3:** es existiert schon ein Knoten $m[x]$, der ein oder mehrere w enthält und für alle anderen Komponenten gleich der Folgemarkierung ist
→ Kante zu diesem Knoten hinzufügen
- **Fall 4:** anderenfalls erhält der Knoten die Bezeichnung der Folgemarkierung.

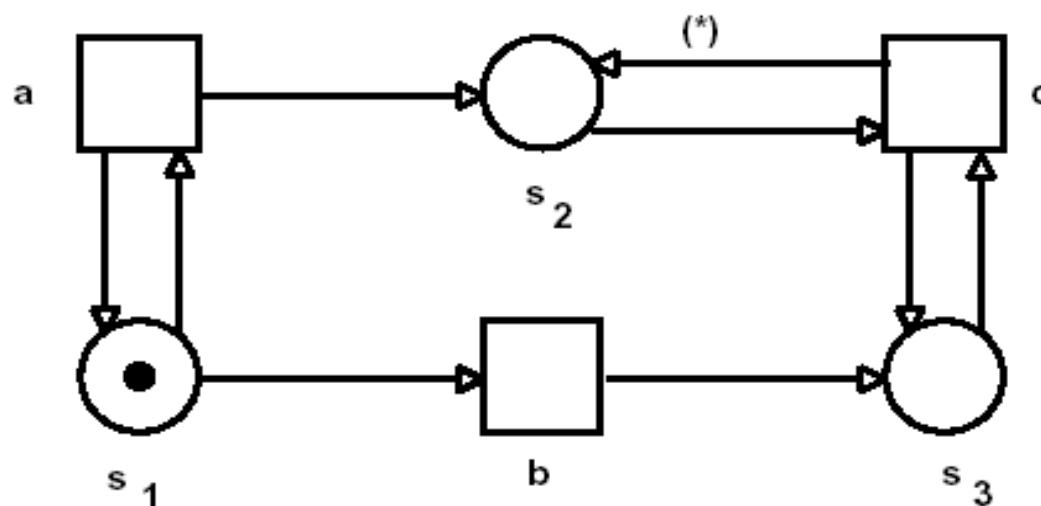
4. Erreichbarkeitsanalyse

- **Überdeckbarkeit** ist ähnlich der Erreichbarkeit,
- eine bestimmte Markierung muss nicht genau erreicht werden, sondern nur eine Teilmenge,
- ist abgeschwächter Begriff der Erreichbarkeit,
- auch partielle Erreichbarkeit,
- nur einzelne Teilzustände betrachtet. Man kann auch durchaus mehrere oder alle beteiligten Zustände überwachen,
- Für ein beschränktes Petrinetz liefert der Überdeckungsgraph den Markierungsgraphen.

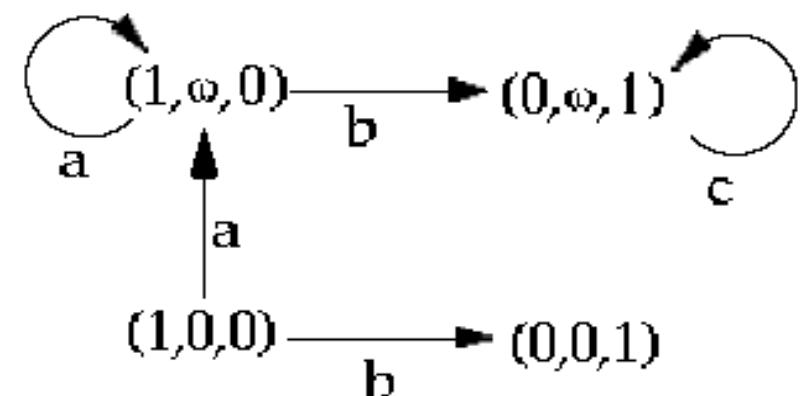
4. Erreichbarkeitsanalyse

Überdeckungsgraphen sind nicht eindeutig:

Verschiedene Netze können denselben Überdeckungsgraphen haben.



Überdeckungsgraph zu N_i ($i = 1,2$):



Sei N_1 mit Pfeil (*), N_2 ohne den Pfeil (*). Beide N_i haben als Überdeckungsgraphen (s.o.).

4. Erreichbarkeitsanalyse

Der Überdeckungsgraph liefert eine notwendige und hinreichende Bedingung für die Fragen:

- Ist eine bestimmte Stelle beschränkt?
- Ist das Petrinetz beschränkt?
- Ist eine bestimmte Transition tot?

Der Überdeckungsgraph liefert eine notwendige Bedingung für die Fragen:

- Ist eine bestimmte Transition lebendig?
- Ist das Petrinetz lebendig?
- Ist eine bestimmte Markierung erreichbar?

Petrinetz als Matrix

- Petrinetze lassen sich in die lineare Algebra abbilden,
- mit deren Hilfe kann man ganze Schaltsequenzen in einem Schritt durchrechnen,
- Matrixdarstellung hilft zur Bestimmung von Stellen-Invarianten und Transitions-Invarianten.

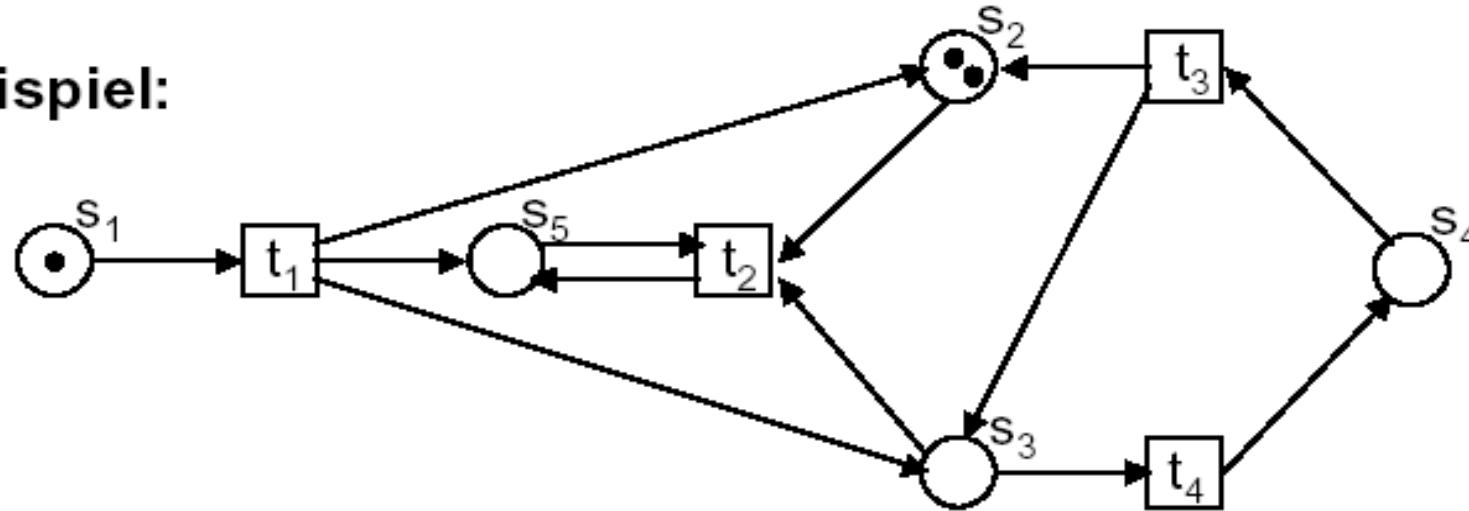
Petrinetz als Matrix

- zu jeder Transition t_i wird definiert:
 - die Abbildung t_i^- zur Transition t_i sagt aus, wie viele Tokens einer Markierung m abgezogen werden, wenn diese schaltet.
 - t_i^+ zu einer Transition t_i sagt aus, wie viele Tokens beim Schalten von t_i einer Markierung m hinzugefügt werden.
 - bei einem Petrinetz mit Transitionen, die jeweils eine Marke pro Stelle erzeugen ergibt sich eine **Inzidenzmatrix I** .

$$I = (i_{jk}) = \begin{cases} -1, & j \in \bullet k \setminus k \bullet, \\ 1, & j \in k \bullet \setminus \bullet k, \\ 0, & \text{sonst} \end{cases} \quad \text{mit } j \in S \text{ und } k \in T.$$

5. Linear-Algebraische Darstellung

Beispiel:



Die **Inzidenzmatrix I** zu obigem Netz lautet:

$$I = \begin{pmatrix} -1 & 0 & 0 & 0 \\ +1 & -1 & +1 & 0 \\ +1 & -1 & +1 & -1 \\ 0 & 0 & -1 & +1 \\ +1 & 0 & 0 & 0 \end{pmatrix}$$

Die Anzahl Zeilen entsprechen der Anzahl Stellen des Netzes und die Anzahl Spalten der Transitionanzahl.

5. Linear-Algebraische Darstellung

Folgemarkierungen

Vektor q_τ zu einer Schaltfolge τ enthält für jede Transition die Anzahl, wie oft sie geschaltet wurde.

z.B. aus der Schaltfolge $\tau = t_1 \ t_4 \ t_3 \ t_4$ folgt der Vektor $q_\tau = (1 \ 0 \ 1 \ 2)$ für die vier Transitionen des Netzes aus dem vorigen Bsp.

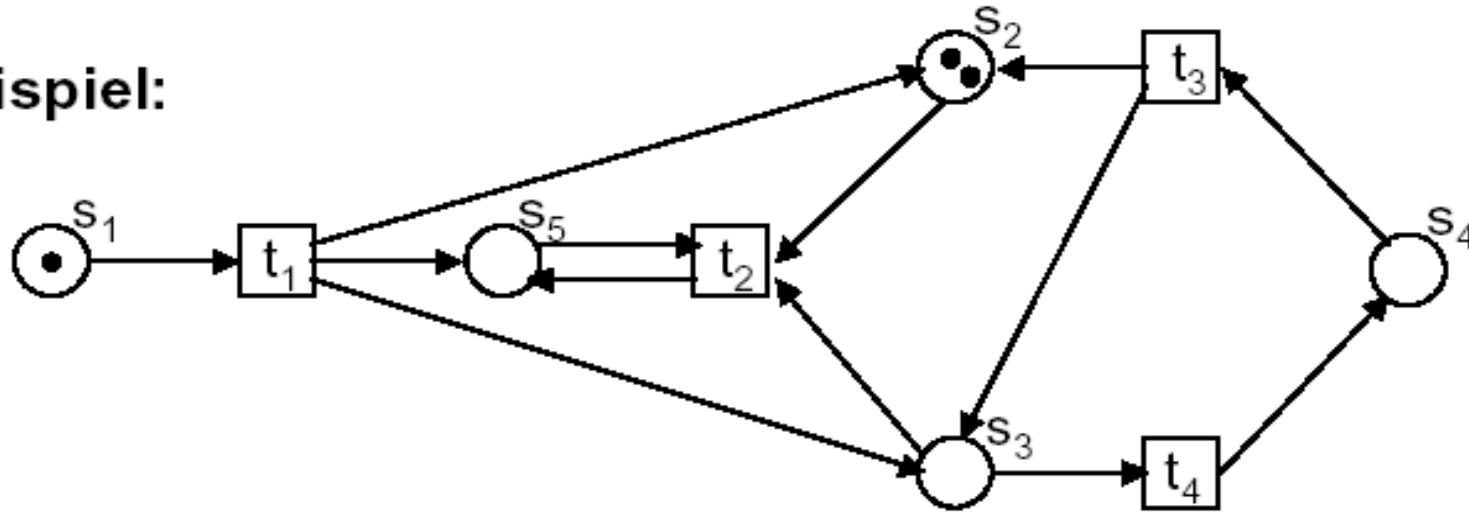
Somit gilt:

Die Markierung m_τ kann von m_0 ausgehend berechnet werden durch:

$$m_\tau = m_0 + (I^* q_\tau^\top)^\top$$

5. Linear-Algebraische Darstellung

Beispiel:



Schaltfolge $\tau = t_1 \ t_4 \ t_3 \ t_4$ \Rightarrow Vektor $q = (1 \ 0 \ 1 \ 2)$

$$m_{\tau} = (1 \ 2 \ 0 \ 0 \ 0) + \left(\begin{pmatrix} -1 & 0 & 0 & 0 \\ +1 & -1 & +1 & 0 \\ +1 & -1 & +1 & -1 \\ 0 & 0 & -1 & +1 \\ +1 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \\ 1 \\ 2 \end{pmatrix} \right)^T = (0 \ 4 \ 0 \ 1 \ 1)$$

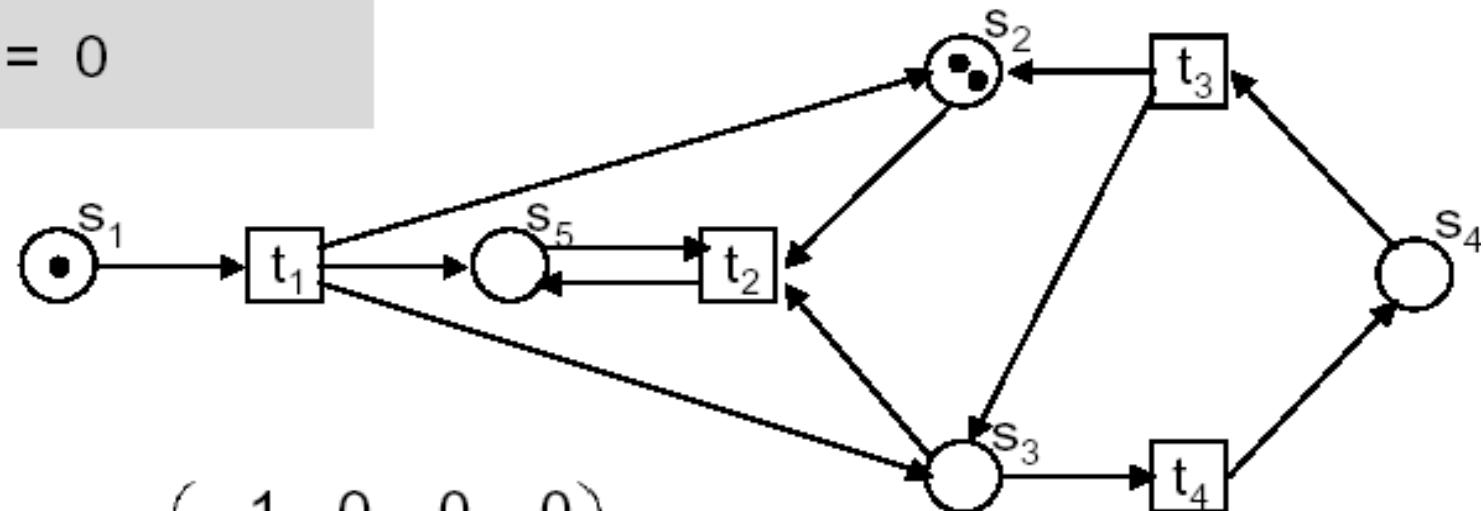
5. Linear-Algebraische Darstellung

Invarianten:

S-Invariante

S-Invarianten $Y = (y_1, \dots, y_{|S|})$ können berechnet werden durch:

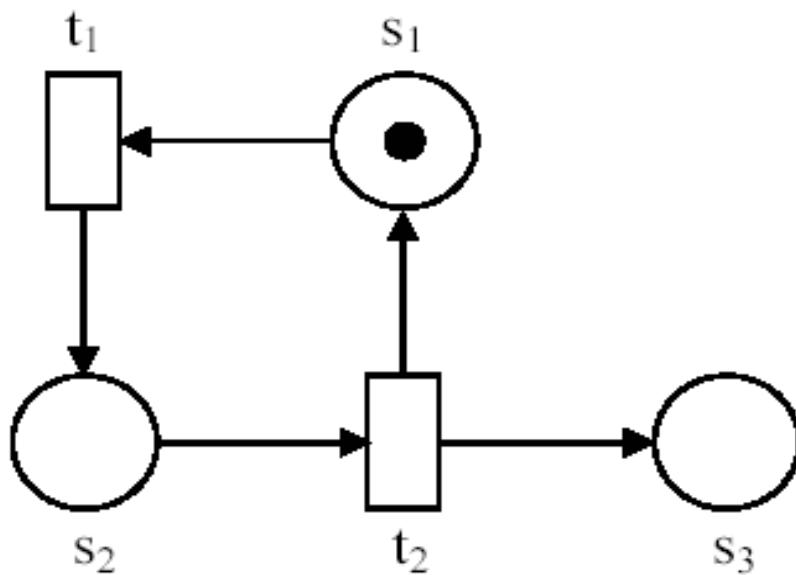
$$Y \cdot I = 0$$



$$(y_1 \ y_2 \ y_3 \ y_4 \ y_5) * \begin{pmatrix} -1 & 0 & 0 & 0 \\ +1 & -1 & +1 & 0 \\ +1 & -1 & +1 & -1 \\ 0 & 0 & -1 & +1 \\ +1 & 0 & 0 & 0 \end{pmatrix} = 0$$

$$Y = (1 \ 0 \ 0 \ 0 \ 1)$$

5. Linear-Algebraische Darstellung



hat die
Inzidenzmatrix ...

$$\begin{matrix} & t_1 & t_2 \\ s_1 & \left(\begin{array}{cc} -1 & 1 \\ 1 & -1 \\ 0 & 1 \end{array} \right) \\ s_2 & \left(\begin{array}{cc} 1 & -1 \\ 1 & 1 \end{array} \right) \\ s_3 & \left(\begin{array}{c} 0 \\ 1 \end{array} \right) \end{matrix}.$$

Das lineare Gleichungssystem $\begin{pmatrix} -1 & 1 \\ 1 & -1 \\ 0 & 1 \end{pmatrix}^T \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, d.h.

$$\begin{aligned} -x_1 + x_2 &= 0 \\ x_1 - x_2 + x_3 &= 0 \end{aligned}, \text{ liefert } \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \text{ als eine S-Invariante}$$

5. Linear-Algebraische Darstellung

Folgerungen aus Stelleninvarianten

Wenn eine S-Invariante Y existiert, deren Komponenten für jede Stelle positiv ist, also $y \geq 1$, ist gezeigt,

- dass das Netz N beschränkt ist, weil die Tokenzahl konstant ist,
- dass jeder Fall irgendwann beendet wird und keine Marken im System zurückbleiben.

Es gilt:

$$m \cdot Y^T = m' \cdot Y^T$$

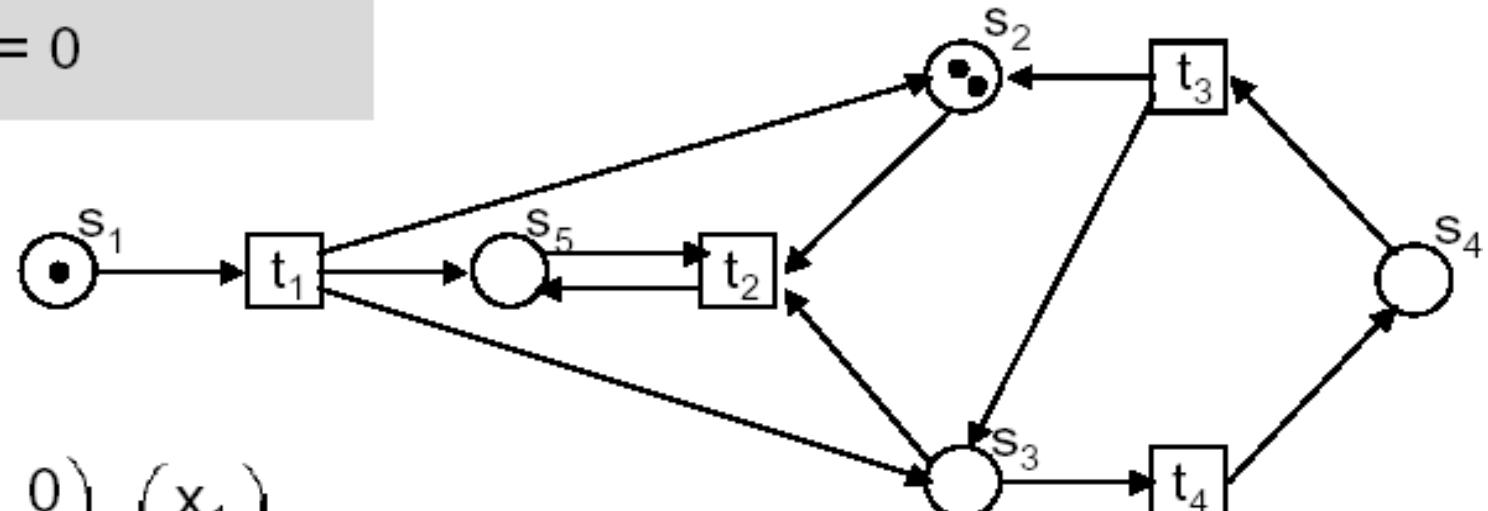
$$(m - m') Y^T = 0$$

5. Linear-Algebraische Darstellung

Invarianten: T-Invariante

T-Invarianten $X = (x_1, \dots, x_{|\Pi|})$ können berechnet werden durch:

$$I \cdot X^T = 0$$

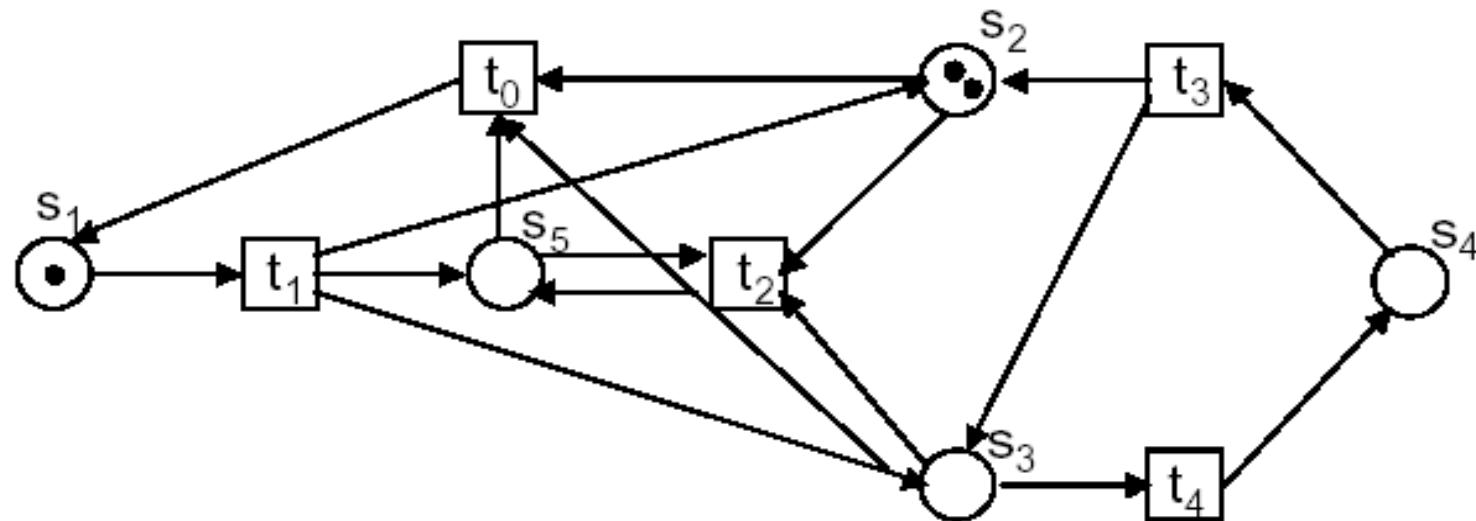


$$\begin{pmatrix} -1 & 0 & 0 & 0 \\ +1 & -1 & +1 & 0 \\ +1 & -1 & +1 & -1 \\ 0 & 0 & -1 & +1 \\ +1 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 0$$

$$X = (0 \ 0 \ 0 \ 0)$$

5. Linear-Algebraische Darstellung

Invarianten: T-Invariante



$$I \cdot X^T = 0$$

$$\begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & +1 & -1 & +1 & 0 \\ -1 & +1 & -1 & +1 & -1 \\ 0 & 0 & 0 & -1 & +1 \\ -1 & +1 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 0$$

$$X = (1 \ 1 \ 0 \ 0 \ 0)$$

Eigenschaften in Sätzen (II)

Ein markiertes Petrinetz ist genau dann verklemmungsfrei, wenn der zugehörige Markierungsgraph keinen Knoten ohne Nachfolger besitzt.

Ein markiertes Petrinetz terminiert genau dann, wenn der zugehörige Markierungsgraph zyklenfrei und endlich ist.

Ein markiertes Petrinetz ist genau dann reversibel, wenn der dazugehörige Markierungsgraph stark zusammenhängend ist.

Falls (N, m_0) beschränkt ist mit Schranke b , dann sind höchstens $(b+1)^{|s|}$ Markierungen erreichbar.
($|s|$ - sei die Anzahl der Stellen im Netz)

6. Zusammenhänge bei der Prozess-Analyse

dynamische Eigenschaft des Petri - Netzes	Interpretation	Eigenschaft des Markierungsgraphen
beschränkt	kein „Überlauf“ von Marken in Stellen	endlich
verklemmungsfrei	kein Stillstand erreichbar	jeder Knoten besitzt einen Nachfolger
reversibel	rücksetzbar in den Anfangszustand	stark zusammenhängend
Markierung M ist erreichbar	ein entsprechender Zustand ist erreichbar	M ist Knoten des Graphen

Geschäftsprozess-Management

Prof. Dr.-Ing. Andreas Ittner

Email: ittner@hs-mittweida.de

WWW: www.hs-mittweida.de/~ittner

Tel.: +49(0)3727-58-1288

Mob.: +49(0)177-5555-347

Fax.: +49(0)180-35518-22467

Gliederung

- Motivation
- Prozesse und Prozess-Management
 - Geschäftsprozesse, Workflow-Prozesse
 - Prozessdesign, Prozessverbesserungen
- Prozess-Modellierung
 - Zweck, Modellierungselemente und –sprachen
 - Petri-Netze, EPKs, BPMN, ...
- Prozess-Analyse
 - Struktur-, Verhaltens-, Erreichbarkeits- und **Performance-**Analysen
 - Simulation
- Workflow-Management-Systeme
 - Historie, Infrastruktur, Implementierungen, Standards

7. Performance-Analyse / Simulation

1. Fragestellungen bei der Prozess-Analyse,
2. Petrinetze mit Zeit,
3. Simulation,
4. Design: Verschiedene Szenarien.

1. Fragestellungen bei der Prozess-Analyse

Wdh.: Arten der Analyse:

- Validierung
 - Ist das Modell richtig bzgl. der Realität/Vorstellung?
 - z.B. Kundenbezug, Medien- und Organisationsbrüche
- Verifikation
 - Nachweis der Korrektheit des Prozesses,
 - Struktur (z.B. Vor- und Nachbedingungen für alle Aktivitäten),
 - Verhalten (z.B. Deadlocks, nie ausgeführte Aktivitäten).
- Leistungsbewertung
 - Leistungsfähigkeit des Geschäftsprozesses,
 - z.B. Durchlaufzeit, Kostenrechnung, Ressourcenauslastung, Service-Level.

Qualitative Fragestellungen

- Kann es zu Deadlocks (toten Markierungen) kommen?
- Können alle Arten von Cases (Fällen) erfolgreich behandelt werden?
- Werden alle Cases irgendwann beendet?
- Können Tasks (Aufgaben) parallel ausgeführt werden?

Quantitative Fragestellungen

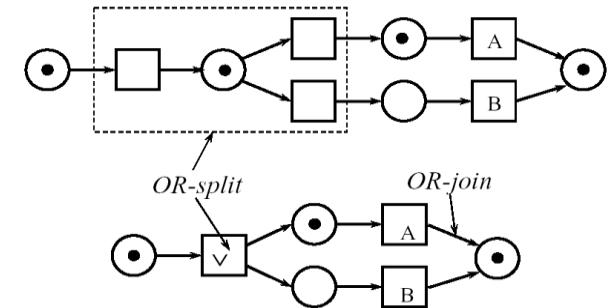
- Wie viele Cases pro Stunde können bearbeitet werden?
- Was ist die durchschnittliche Durchlaufzeit?
- Wie viele Ressourcen werden pro Case benötigt?
- Was ist die durchschnittliche Wartezeit?

2. Petrinetze mit Zeit

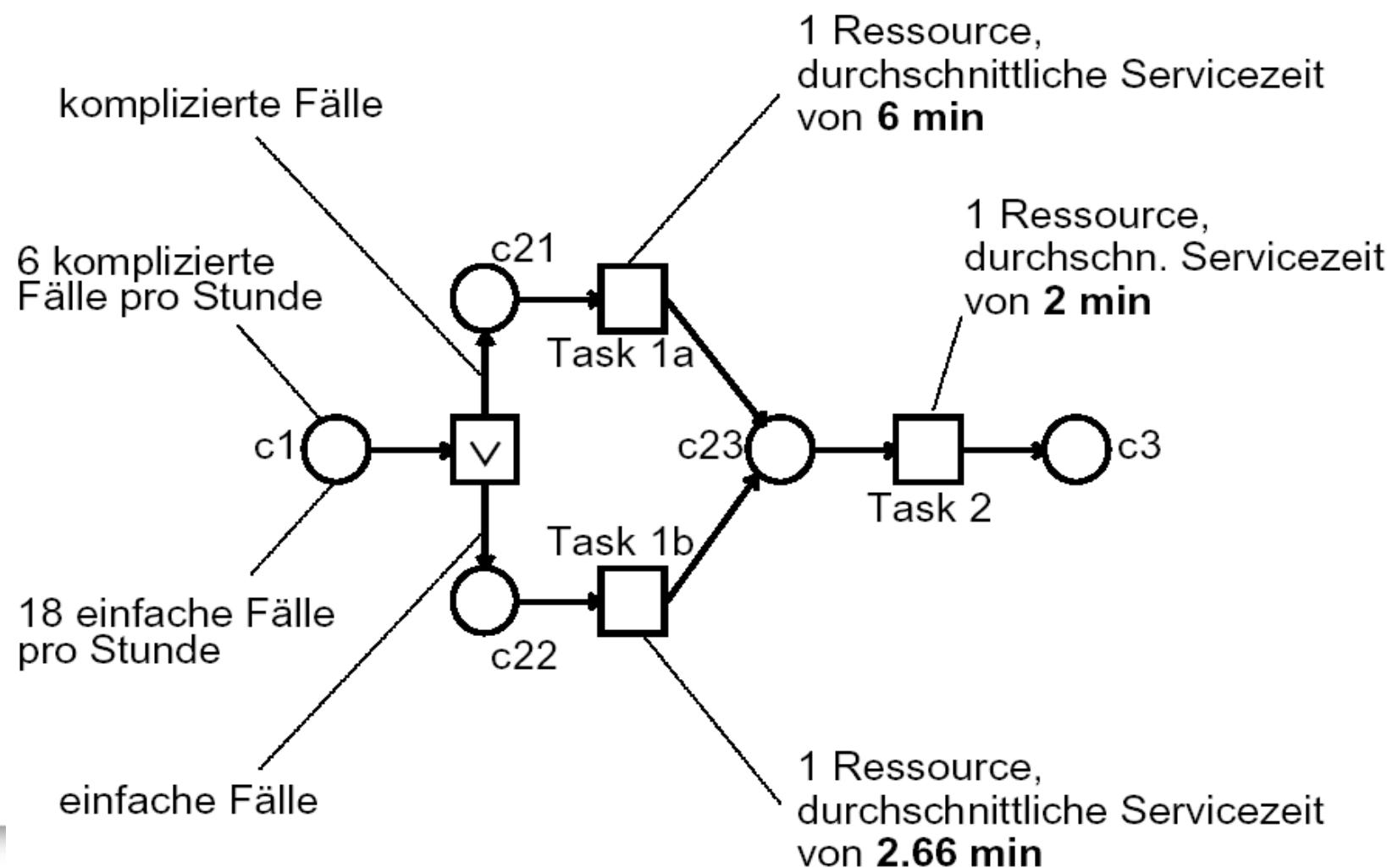
Die Erweiterung mit Zeit:

- für Performance-Analysen werden Zeitdauern, Verzögerungen etc. benötigt,
- jeder Transition t wird eine Zeitdauer $\delta(t)$ zugeordnet
 - d.h. zuerst wird t initialisiert und die Marke verschwindet von der Inputstelle s ,
 - dann ist t aktiv während der Zeitdauer $\delta(t)$,
 - Beendigung der Transition t , die Marke erscheint in der Outputstelle.
- es können für $\delta(t)$ Zufallsverteilungen angenommen werden, da z.B. die Ankunft der Kunden und die Dauer des Service i.a. nicht bekannt sind.

7. Performance-Analyse / Simulation

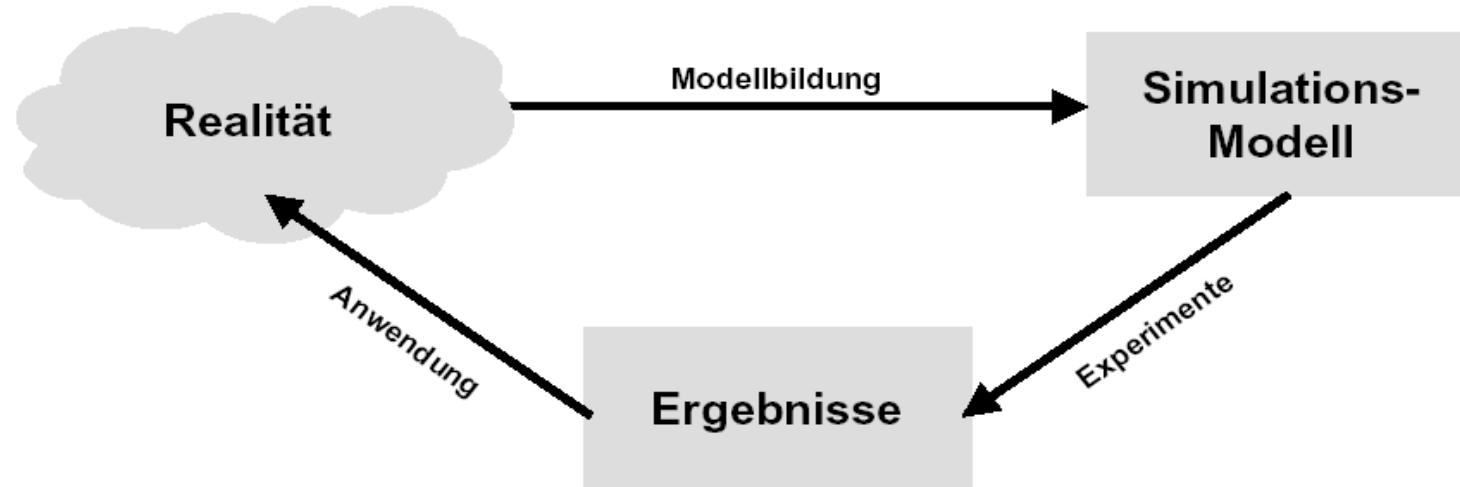


2. Petrinetze mit Zeit (Beispiel)



3. Simulation

- ist das systematische Experimentieren mit Modellen realer Problemstellungen.



- Prozesse realer Systeme können anhand von Modellen durchgeführt und ausgewertet (simuliert) werden
- Ergebnisse der Simulationsexperimente werden dann auf das reale System übertragen.

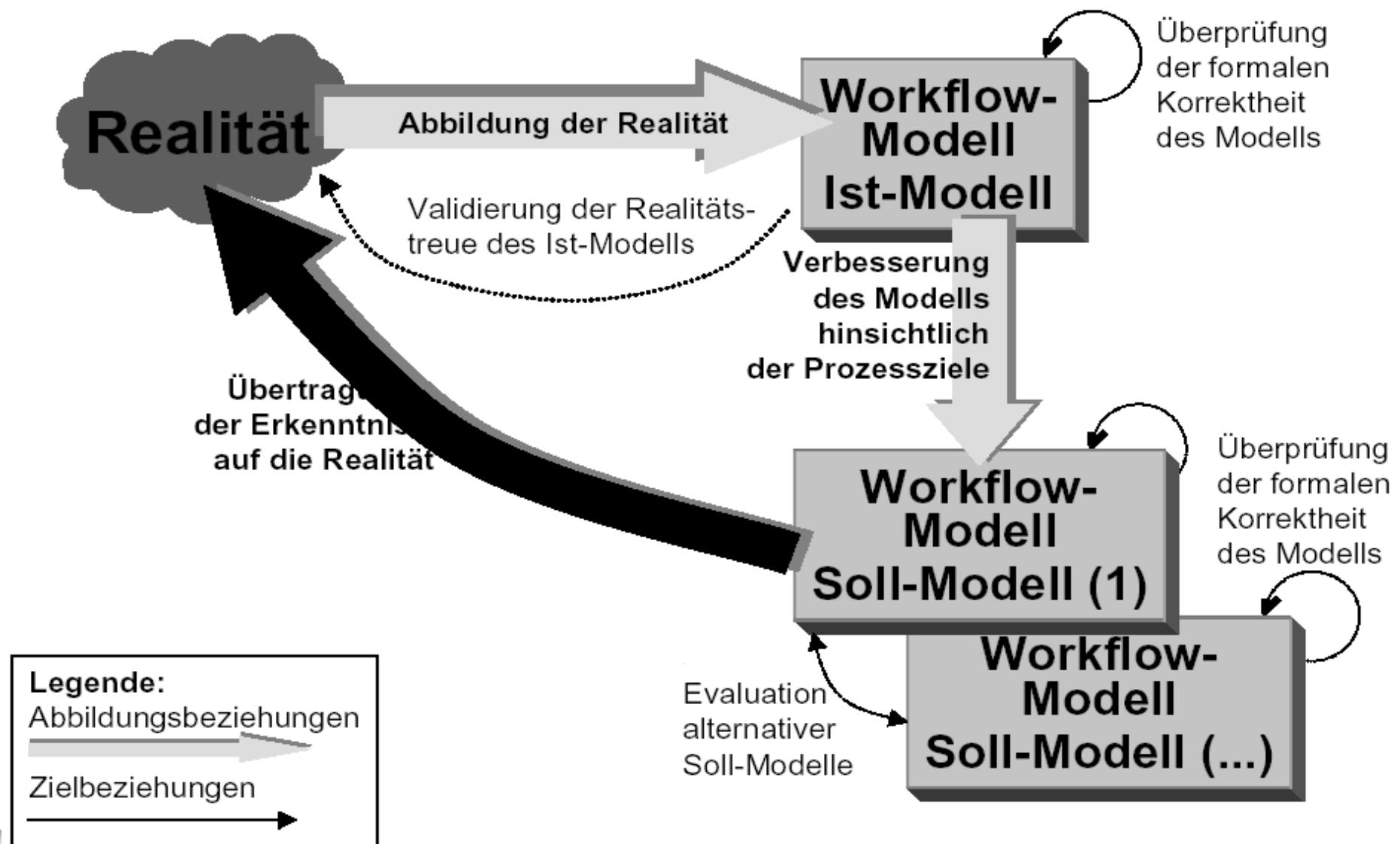
3. Simulation

- Simulation = Probebetrieb,
 - in der Regel Computerexperiment unter Verwendung von Pseudozufallszahlen,
 - kann notwendiger Ersatz für Realexperimente sein (unwirtschaftlich, unmöglich),
 - Hilfsmittel, um die Komplexität der Realität zu vereinfachen,
 - erlaubt Blick auf wesentliche Probleme,
 - ist Instrument zur qualitativen Verbesserung und Absicherung der Unternehmensplanung,
 - ist Instrument zur Entscheidungsunterstützung.
- Kreativität zur Lösungsfindung bleibt aber beim Anwender !

3. Simulation / Vorteile der Simulation

- Visualisierung von Prozessen,
- Verfolgung von Kosten und Wertschöpfung,
- Qualitätskontrolle der Modelle,
- Risikoreduzierung,
- Erhöhung der Planungsqualität,
- Zeit- und Kostensparnis,
- Vergleich mehrerer Alternativen,
- Vermeidung von Betriebsunterbrechungen.

3. Simulation / Ziele der Workflow-Simulation



3. Simulation / Ziele der Workflow-Simulation

- Überprüfung der Ablauffähigkeit von Workflow-Modellen
 - Überprüfung auf formale Korrektheit und Konsistenz,
 - noch keine Aussagen über den Inhalt der Modelle,
- Validierung der Realitätstreue von Workflow-Modellen
 - fachlich-inhaltliche Korrektheit,
 - inwieweit bildet das Workflow-Modell die Realität angemessen ab,
 - Validierung des Inhaltes des Modells (Gegenüberstellung von relevanten Ist-Daten der Realität und Simulationsergebnissen),
- Evaluation alternativer Workflow-Modelle
 - entscheidungsunterstützende Informationen für die qualitative Verbesserung der Workflow-Modelle und somit auch der Abläufe,
 - Vergleich von Simulationsergebnissen und Zielkennzahlen (wie mittlere Durchlaufzeiten, Kapazitätsauslastungen, Prozesskosten)
 - „Was wäre wenn“ Analysen.

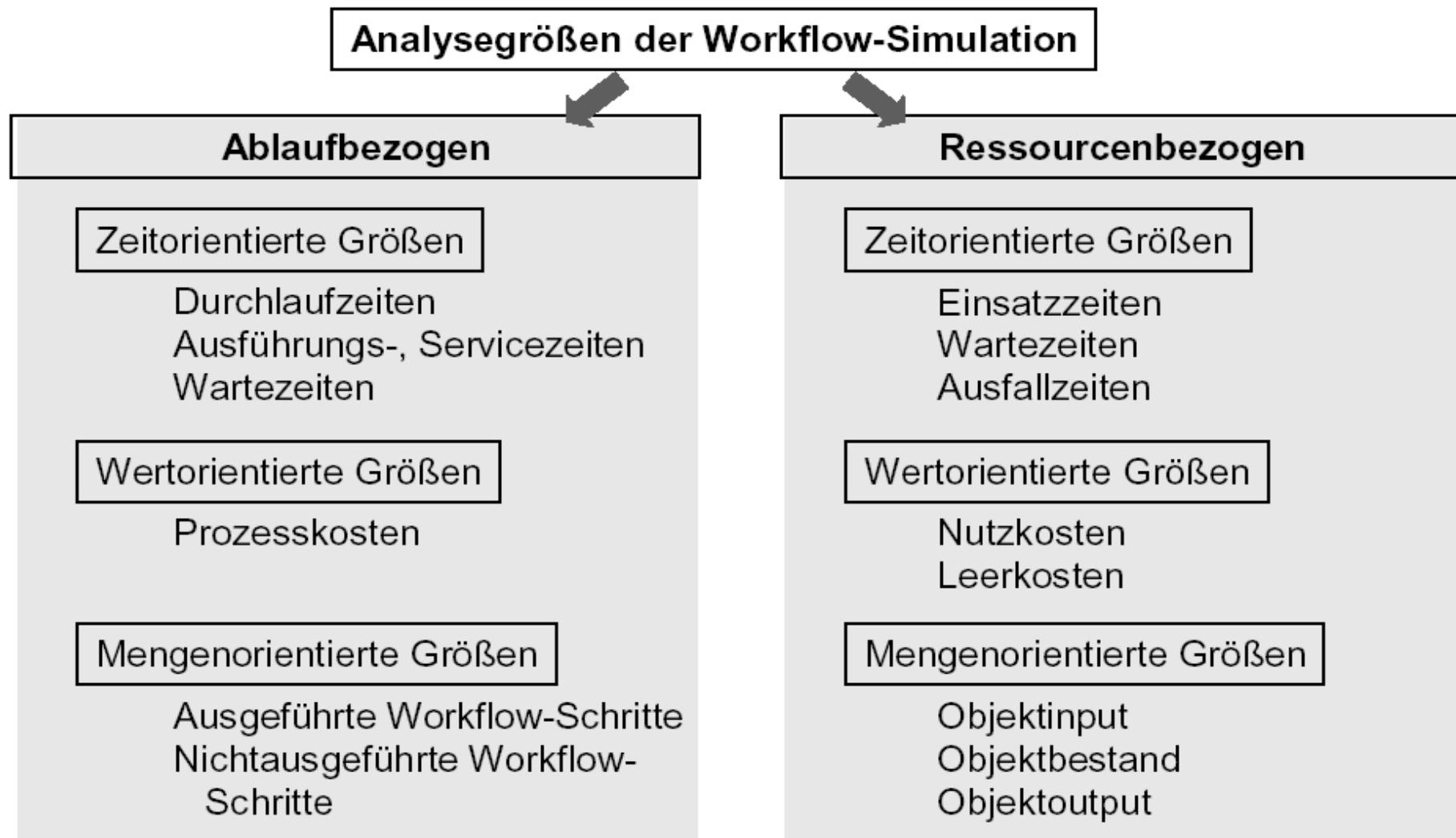
3. Simulation / Analysegrößen der Workflow-Simulation

- Unterteilung in
 - ablaufbezogene und
 - ressourcenbezogene Analysegrößen

(Betrachtung der im Rahmen der Simulation erzeugten Workflowinstanzen hinsichtlich des Ablaufverhaltens bzw. der Ressourcen (Bearbeiter, Programme,...)).
- weitere Differenzierung in
 - zeit-,
 - wert- und
 - mengenorientierte Größen.

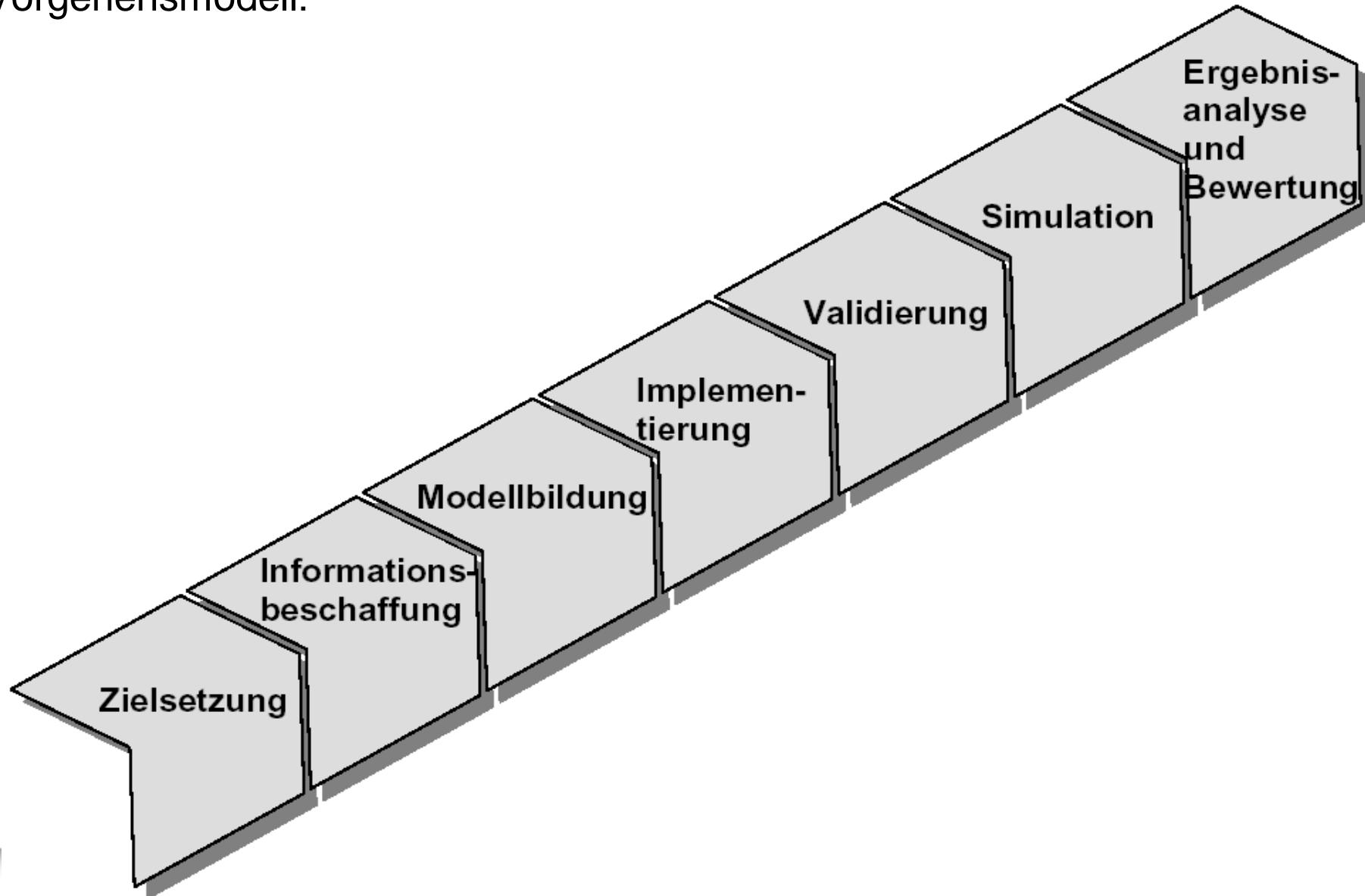
3. Simulation / Analysegrößen der Workflow-Simulation

nach Gadatsch



3. Simulation / Durchführung der Simulation - 7 Schritte

Vorgehensmodell:



3. Simulation / Durchführung der Simulation - 7 Schritte

1. Zielsetzung festlegen

- vor Beginn müssen die Ziele quantitativ, also rechnerisch nachvollziehbar, schriftlich festgelegt/fixiert werden,
Beispiel: Reduzierung der Durchlaufzeiten um ...%,

2. Informationsbeschaffung

- Erfassung aller relevanten Basisdaten für die Simulationsuntersuchung einschließlich einer Plausibilitäts- und Vollständigkeitsprüfung,
Beispiel: Bearbeitungszeiten, Kapazitäten, Störgrößen,
- Daten müssen klassifiziert und verdichtet werden, so dass sie den Aufgaben zugeordnet werden können.

3. Modellbildung

- Festlegung „was mit den Daten geschehen soll“,
Beispiel: Bearbeitungsreihenfolgen, Prioritätsregeln

3. Simulation / Durchführung der Simulation - 7 Schritte

4. Implementierung des Modells

- konkrete Erfassung des Modells im Simulator,
- Detaillierungsgrad abhängig von der fixierten Zielsetzung,
- Modellbildung und Implementierung können evtl. gleichzeitig erfolgen, da viele Werkzeuge eine rechnergestützte Modellierung anbieten,

5. Überprüfung des Modells (Validierung)

- Überprüfung der Übereinstimmung mit der abzubildenden Realität,
Beispiel: Vergleich von Vorab-Simulationsläufen mit bekannten Ergebnissen
(evtl. Daten des Vorjahres),

6. Experimentieren mit dem Modell (Simulation)

- Simulationsphase: verschiedene Versuchsreihen mit variierenden Parametern,
- Dokumentation nicht vergessen, die jeden Parameterwechsel protokolliert!
- Beispiel: unterschiedliche Planungszeiträume (Woche, Jahr,...), unterschiedliche Kapazitäten.

3. Simulation / Durchführung der Simulation - 7 Schritte

7. Ergebnisanalyse und Bewertung

- numerische Ergebnisse können graphisch dargestellt und analysiert werden (Diagramme, etc.),
- Auswertung kann zu Modelländerungen führen
- → neue Simulationsdurchführung (Beginn bei Schritt 1 oder später wenn möglich).

3. Simulation / Durchführung der Simulation – weitere Tipps:

- Überschaubarkeit/Übersichtlichkeit bewahren
(keine zu große Anzahl von Experimenten)
- pro Simulationslauf nicht mehrere Parameter gleichzeitig ändern !!!
(sonst sind Ursache und Auswirkung nicht direkt erkennbar)

3. Simulation / Warteschlangenmodell:

- ist dafür geeignet, eine Leistungsanalyse von Systemen oder aber auch von Geschäftsprozessen durchzuführen,
- Warteschlangen sind charakterisiert durch Personen, die
 - unregelmäßig ankommen,
 - und auf ihre Abfertigung (Service) oder ihren Durchlass warten.
- es gibt einen oder mehrere Server (also Bediener), die den Service anbieten,
- die Ankunftszeitpunkte der Personen sowie die Dauer des Services sind i. Allg. nicht bekannt (Annahme von Zufallsverteilungen),
- Modelle können auch statt auf Personen auf allgemein auch auf zu bearbeitende Fälle, Ausfälle bei Maschinen/Ressourcen, etc. angewendet werden.

7. Performance-Analyse / Simulation

3. Simulation / Warteschlangenmodell:

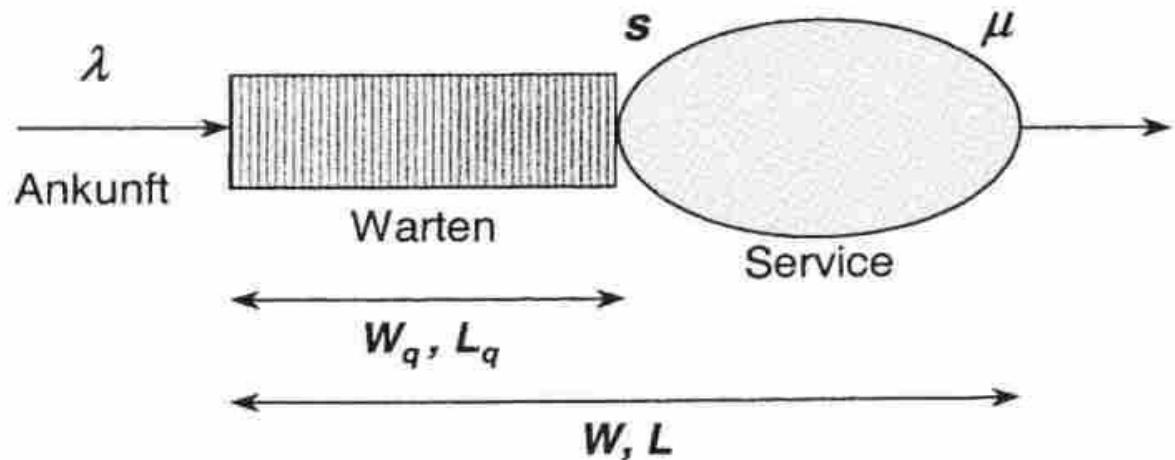
Beispiele:

Kunde	Bedienstation	Warteschlange
Zwischenprodukte	Maschine	Puffer vor der Maschine
Skifahrer	Ski-Lift	Ski-Fahrer vor dem Ski-Lift
Flugzeuge	Startbahn	Flugzeuge auf dem Rollfeld
Flugzeuge	Landebahn	Flugzeuge in der Warteschleife
Touristen	Airline-Schalter	Touristen vor dem Schalter
Reisende	Fahrkartenschalter/-automat	Reisende vor dem Schalter
Kunden	Supermarkt-Kasse	Kundenschlange
Fahrzeuge	Tankstelle oder Ampel	Fahrzeugschlange
Patienten	Arzt	Wartezimmer
Bestellungen	Verkaufsabteilung	Lieferverpflichtungen
Druckaufträge	Drucker	Druck-Jobs
Studenten	Mensa	Studenten vor der Essensausgabe
Maschinen	Instandhaltung	Reparaturaufträge
Transportgüter	Transportsystem	Zwischenlager
Lastwagen	Laderampe	Kolonne
Anrufe	Vermittlungsstelle	Anrufwiederholer
Prozesse	CPU	Prozess-Warteschlange



7. Performance-Analyse / Simulation

3. Simulation / Warteschlangenmodell:



μ – Anzahl Fälle, die pro Zeiteinheit abgearbeitet werden können (Bedienrate),

λ – durchschnittl. eintreffende Fälle pro Zeiteinheit (Ankunftsrate),

s – Anzahl der Server, die den Service durchführen,

W_q – durchschnittl. Wartezeit in der Warteschlange,

L_q – durchschnittl. Anzahl von Fällen in der Warteschlange,

L – durchschnittl. Anzahl von Fällen im System,

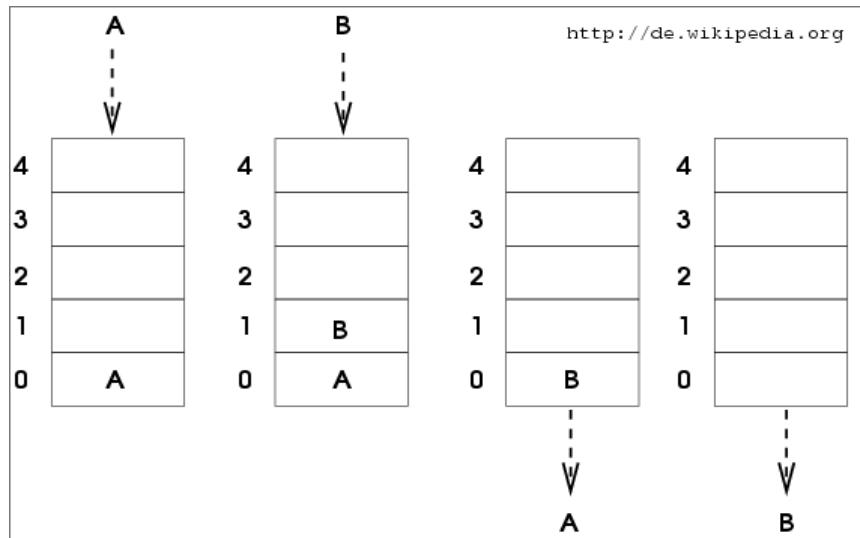
W - durchschnittl. Verweildauer im System.

3. Simulation / Warteschlangenmodell:

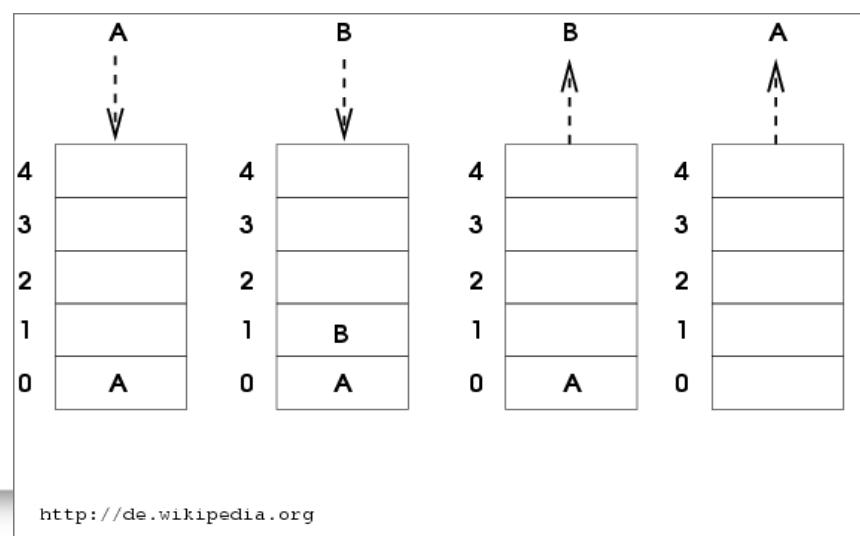
- **Einfachstes** Warteschlangenmodell mit nur einem Server, dem FIFO/FCFS-Prinzip, exponentialverteilten Zwischenankunfts- und Bedienzeiten (Poissonverteilung der Ankunfts- und Bedienrate):
 - μ – Mittlere Bedienrate,
 - $1/\mu$ – Mittlere Bedienzeit,
 - λ – Mittlere Ankunftsrate,
 - $1/\lambda$ – Mittlere Zwischenankunftszeit,
 - $r = \lambda / \mu$ – Mittlere Auslastung,
 - $W = 1 / (\mu - \lambda)$ – Mittlere Durchlaufzeit,
 - $W_q = \lambda / (\mu * (\mu - \lambda))$ – Mittlere Wartezeit,
 - $L_q = \lambda^2 / \mu * (\mu - \lambda)$ – Mittlere Anzahl von Kunden/Fällen in der Warteschlange,
 - $L = \lambda / (\mu - \lambda)$ – Mittlere Anzahl von Kunden/Fällen im System.

7. Performance-Analyse / Simulation

3. Simulation / Warteschlangenmodell: FIFO:

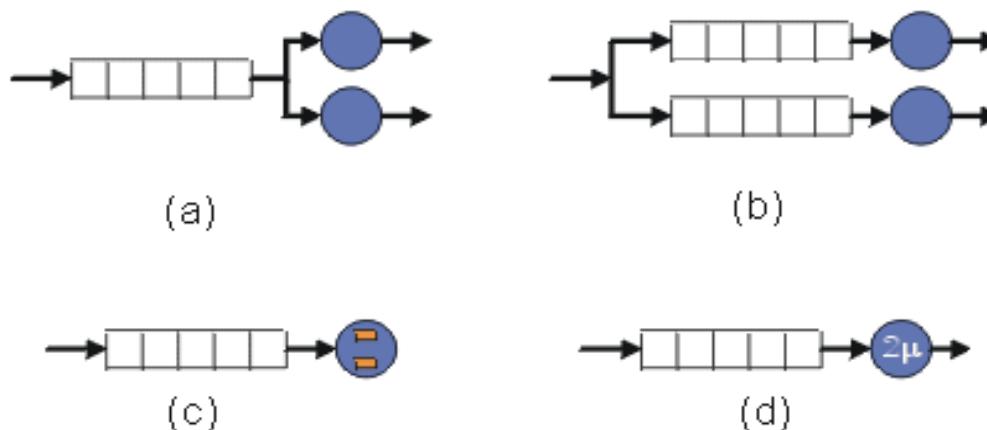


LIFO:



3. Simulation / Warteschlangenmodell:

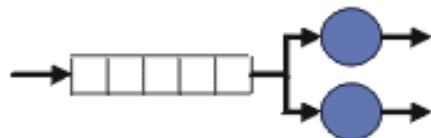
Design von Warteschlangensystemen (exemplarisch 4 typische Muster):



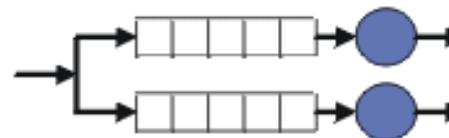
- a) Es stehen zwei Bediener zur Verfügung. Die ankommenden Kunden werden von einem so genannten Dispatcher dem jeweils freiwerdenden Schalter zugewiesen. Nach diesem Prinzip werden z. B. beim Check-In auf dem Flughafen die Warteschlangen abgearbeitet.
- b) Auch hier stehen zwei parallele Bediener zur Verfügung. Allerdings werden die Kunden bereits bei ihrer Ankunft gleichmäßig (d. h. zu je 50%) auf die beiden Warteschlangen verteilt. In dieser Weise stellt man sich an den Fahrkartenschaltern auf Deutschlands Bahnhöfen an.

3. Simulation / Warteschlangenmodell:

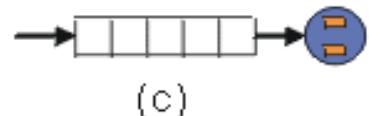
Design von Warteschlangensystemen (exemplarisch 4 typische Muster):



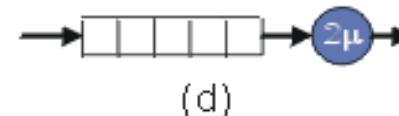
(a)



(b)



(c)



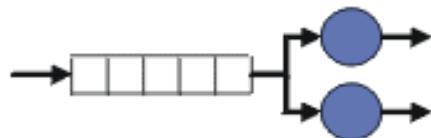
(d)

- c) Es gibt nur einen Bediener und eine Warteschlange. Der Bediener kann jedoch zwei Kunden gleichzeitig bedienen.
- d) Es gibt nur einen Bediener und eine Warteschlange. Der Bediener arbeitet jedoch mit doppelter Geschwindigkeit.

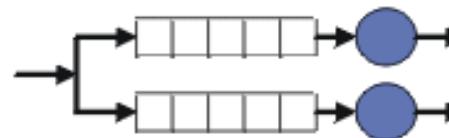
7. Performance-Analyse / Simulation

3. Simulation / Warteschlangenmodell:

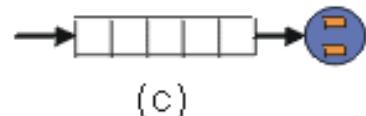
Design von Warteschlangensystemen (exemplarisch 4 typische Muster):



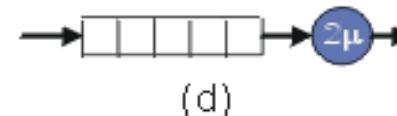
(a)



(b)



(c)

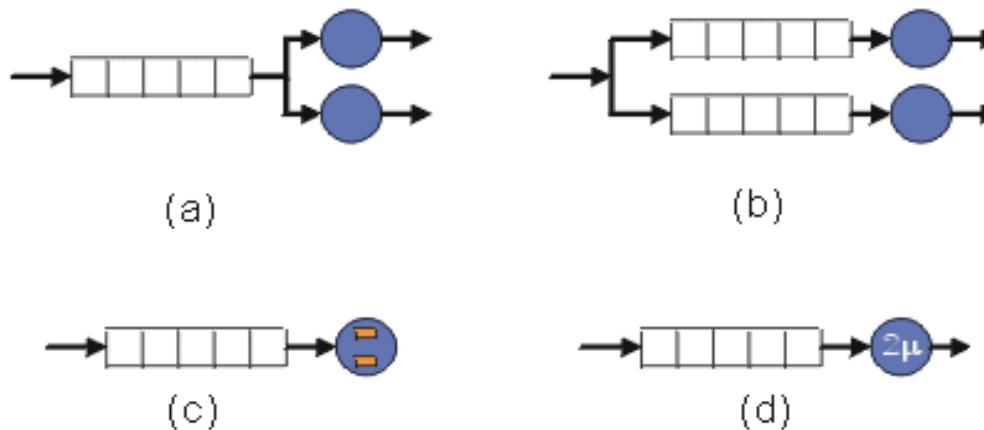


(d)

- Aufgabe: Es stellt sich die Frage, welches dieser Systeme in der Praxis am günstigsten ist?
- Zum Vergleich der Systeme verwenden wir als Leistungsmerkmal die mittlere Anzahl Kunden im System (L).
- Für den Vergleich der Leistungsgrößen werden folgende Parameter angenommen:
Ankunftsrate = 1.8, Bedienrate = 1.0, Variationskoeffizient des Ankunftsstroms $cl = 1.0$, Variationskoeffizient des Bedienprozesses $cs = 0$.

3. Simulation / Warteschlangenmodell:

Design von Warteschlangensystemen (exemplarisch 4 typische Muster):

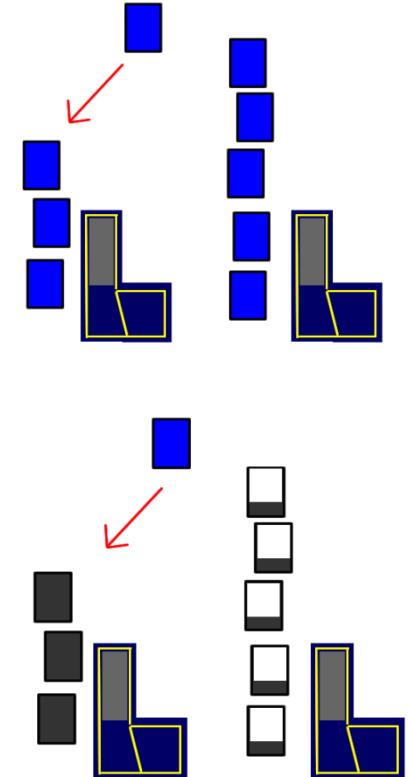


- Lösung: (a) $L = 5,64$, (b) $L = 9,90$, (c) $L = 6,35$, (d) $L = 4,95$
- Das Beispiel zeigt, dass eine gemeinsame Warteschlange günstiger ist als zwei getrennte und dass eine doppelt so schnelle Maschine besser geeignet ist als zwei parallele, halb so schnelle.
- Andererseits leisten zwei parallele Maschinen jedoch mehr als eine einzige Maschine, die zwei Teile gleichzeitig bearbeiten kann.

3. Simulation / Warteschlangenmodell:

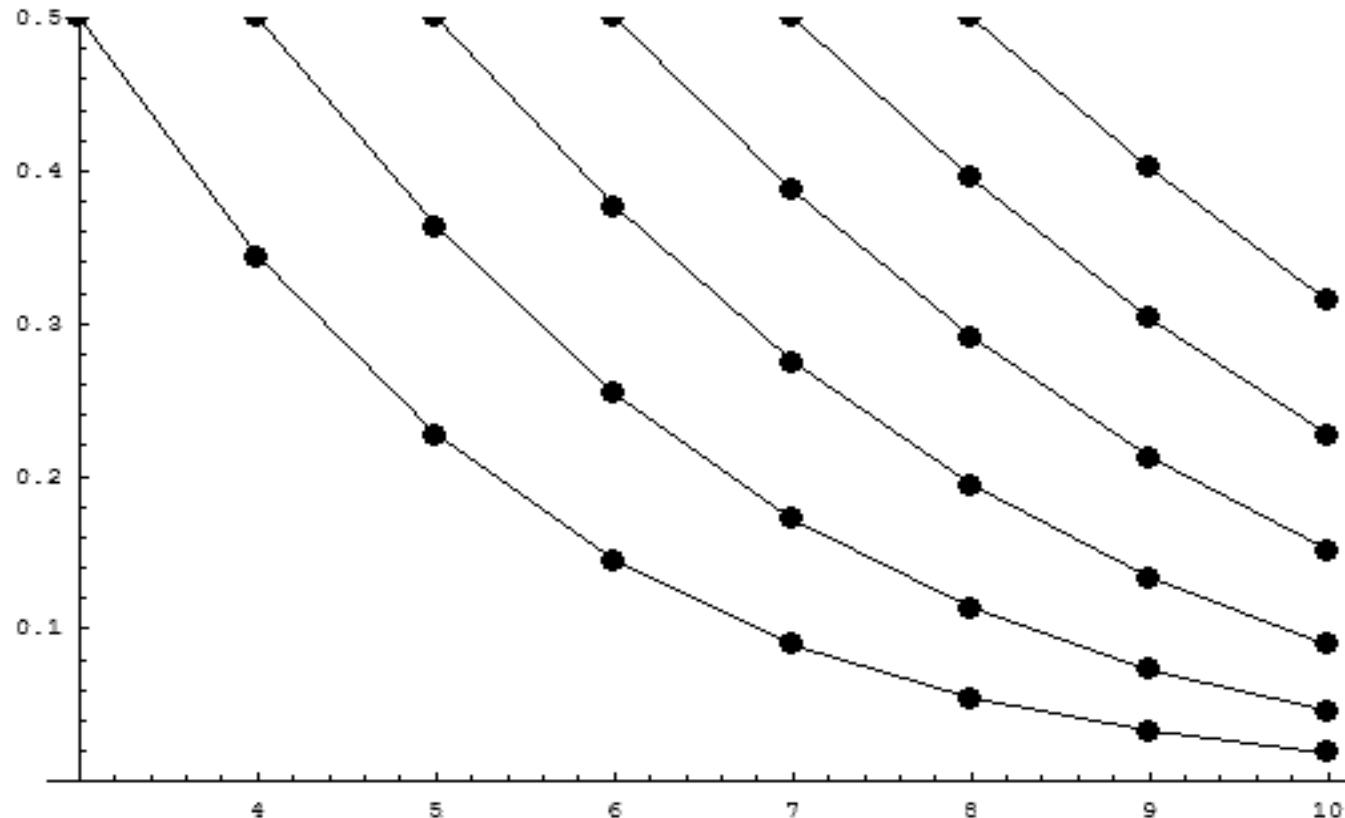
Warum steht man so oft in der falschen Warteschlange ?

- Stehen wie im Supermarkt mehrere Kassen zur Auswahl, so neigt man dazu, sich bei der Kasse mit der kürzesten Warteschlange anzustellen, um die eigene Wartezeit zu minimieren.
- **Problem:** offensichtlich garantiert diese Strategie nicht, dass man auch schneller abgefertigt wird. Da die Bediendauern der einzelnen Kunden zufällig schwanken (der eine hat mehr im Einkaufskorb, der andere weniger), kann es vorkommen, dass in der langen Warteschlange zufällig viele kleine Aufträge aufeinander folgen, während in der kürzeren Schlange große Aufträge vorherrschen. In diesem Fall muss man in der kürzeren Schlange möglicherweise länger warten als in der längeren Schlange.



7. Performance-Analyse / Simulation

3. Simulation / Warteschlangenmodell:



- x-Achse ist die Anzahl der Kunden in der jeweils längeren Warteschlange und auf der y-Achse die Wahrscheinlichkeit, länger warten zu müssen, aufgetragen. Die unterste Kurve steht für "3 Kunden in der eigenen Warteschlange" und die oberste für "8" Kunden in der eigenen Warteschlange".
- Der am weitesten links gelegene Punkt jeder Linie liegt bei einer Wahrscheinlichkeit von 50%, was auch plausibel ist: Denn wenn sich 3 Kunden in der eigenen Warteschlange und 3 Kunden in der anderen Warteschlange befinden, so beträgt die Wahrscheinlichkeit, dass man länger warten muss, gerade 50%.

7. Performance-Analyse / Simulation

3. Simulation / Warteschlangenmodell:

Java Applet unter

<http://www.stochastik.tu-clausthal.de/Presse/Schulen/Rechner>

Rechner für einfache Warteschlangensysteme

Mit diesem Rechner können Sie einfache Wartesituationen analysieren:
(Bitte etwas Geduld, bis das Java-Applet geladen ist)

Ankunftsstrom <input type="text" value="0,9"/> Ankunftsrate (Teile/ZE) Variationskoeffizient 1 <input type="range" value="1"/>	Bedienungsprozess <input type="text" value="1"/> Bedienrate (Teile/ZE) Variationskoeffizient 1 <input type="range" value="1"/> <input type="text" value="1"/> Batchgröße <input type="text" value="1"/> Anzahl paralleler Bediener	Ausfallverhalten Verfügbarkeit: 1 <input type="range" value="1"/> Mittlere Ausfallzeit (ZE) <input type="text" value="0"/> Variationskoeffizient 1 <input type="range" value="1"/>
Leistungsgrößen des Warteschlangenmodells		
Auslastung $\rho = 0,9$	Kunden im System $E[N]=9$ $E[Q]=8,1$ $E[B]=0,9$	Verweilzeit (ZE) $E[V]=10$ $E[W]=9$ $E[C]=1$

4. Design: Verschiedene Szenarien

Bei komplexen Problemen ist es schwierig, eine korrekte Leistungsbewertung durchzuführen: → Simulation

Gegeben:

- 2 Tasks sind pro Kunde zu bearbeiten
- 24 Kunden pro Stunde

Frage:

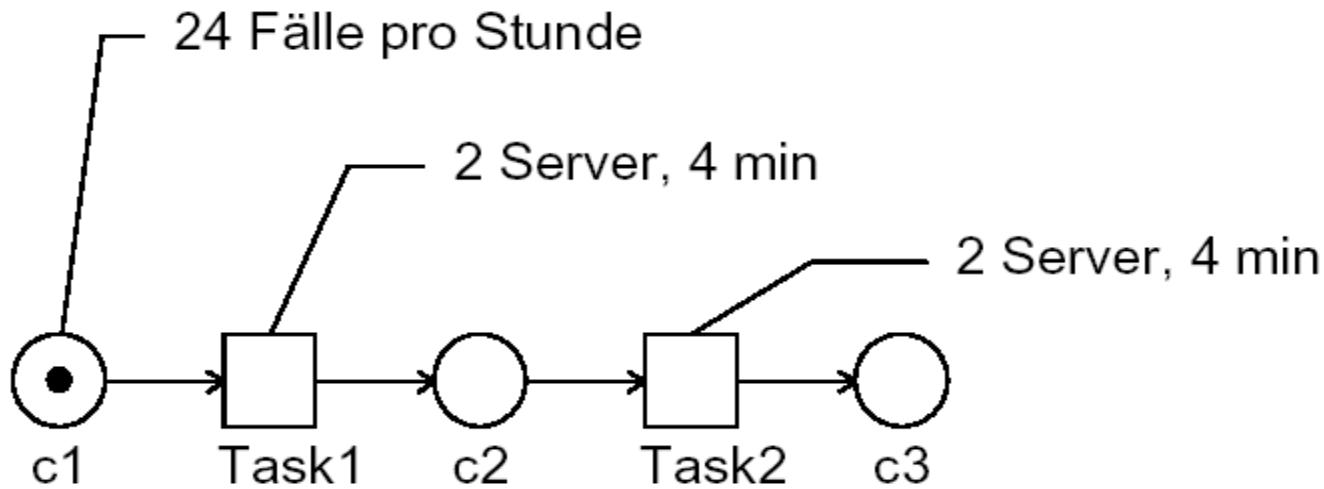
- Wie sieht das beste Design aus?

Zu berechnen:

- durchschnittliche Durchlaufzeit,
- durchschnittliche Servicezeit,
- durchschnittliche Wartezeit,
- (Auslastungsgrade,)
- (durchschnittliche Anzahl im System.)

7. Performance-Analyse / Simulation

4. Szenario 1: sequentiell



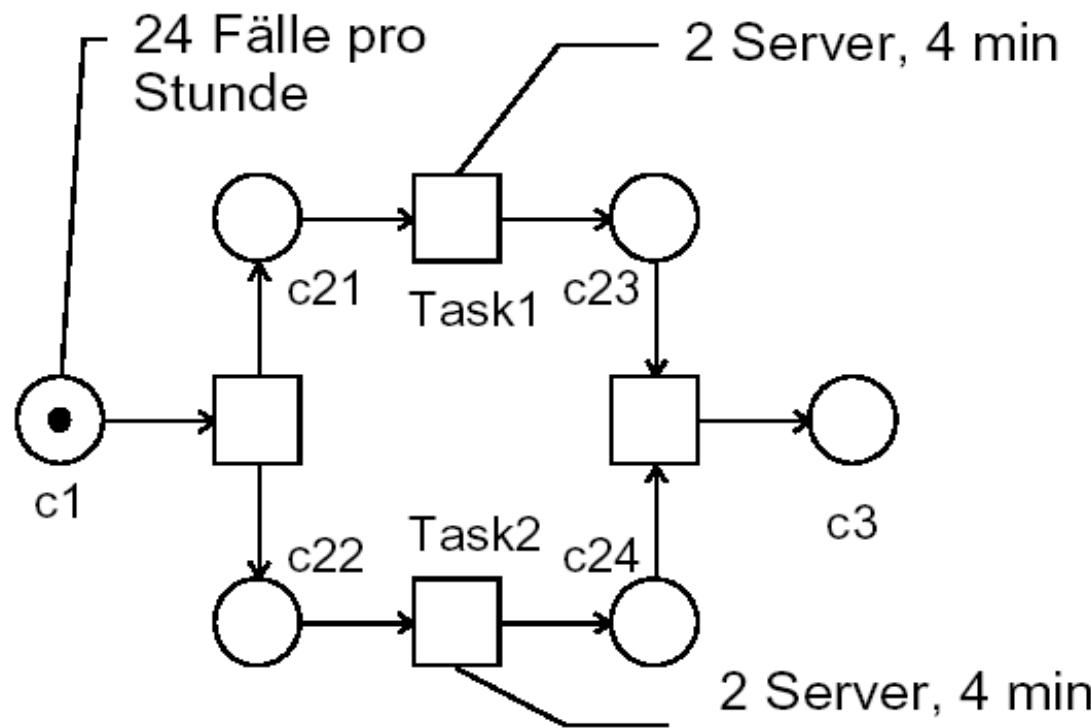
Durchlaufzeit : 22.2 min

Servicezeit: 8.0 min

Wartezeit: 14.2 min

7. Performance-Analyse / Simulation

4. Szenario 2: parallel

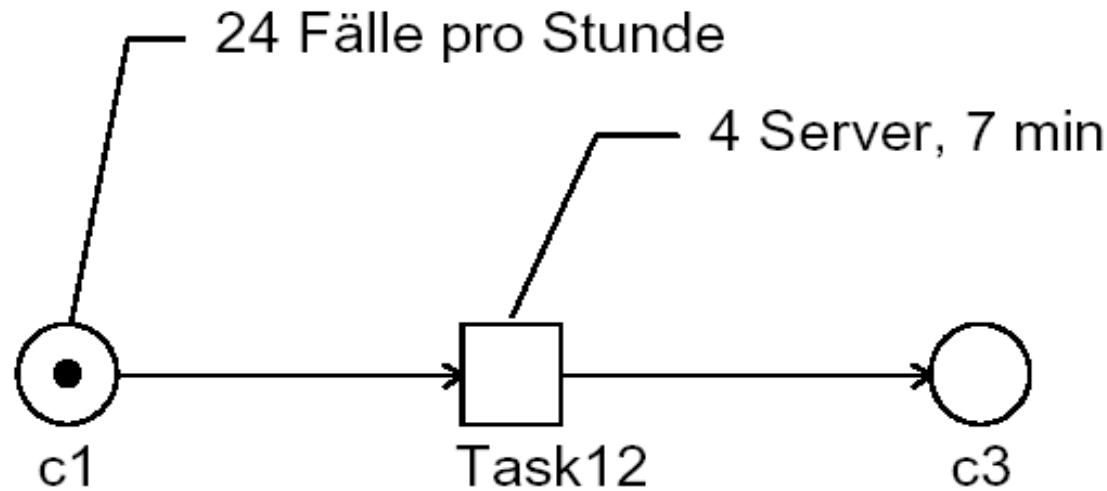


Durchlaufzeit : 15.0 min

Servicezeit: 4.0 min

Wartezeit: 11.0 min

4. Szenario 3: Komposition

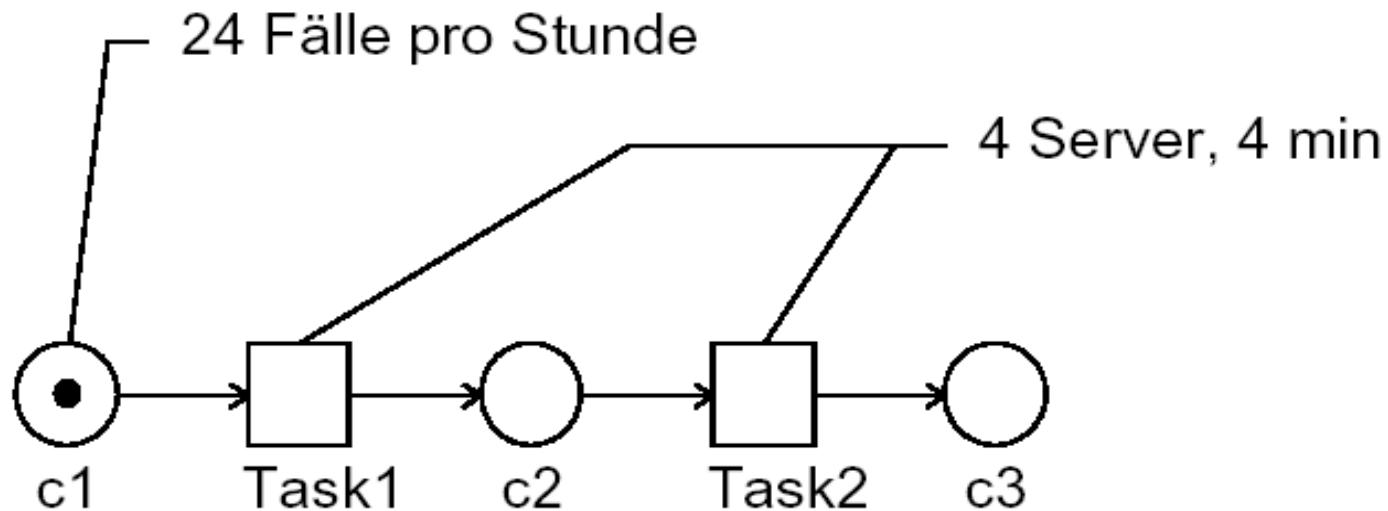


Durchlaufzeit : 9.5 min

Servicezeit: 7.0 min

Wartezeit: 2.5 min

4. Szenario 4: flexible Ressourcen



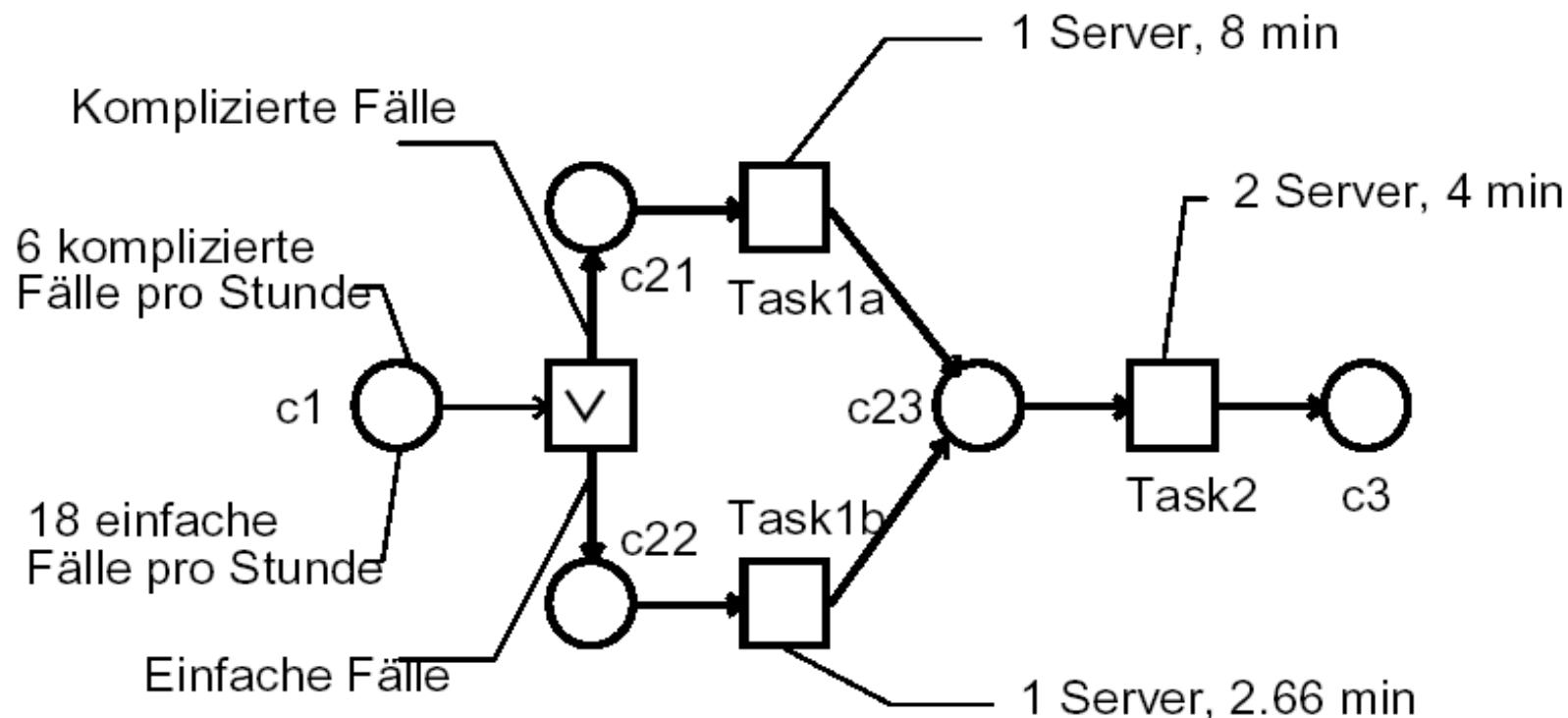
Durchlaufzeit : 14.0 min

Servicezeit: 8.0 min

Wartezeit: 6.0 min

7. Performance-Analyse / Simulation

4. Szenario 5: Vorauswahl



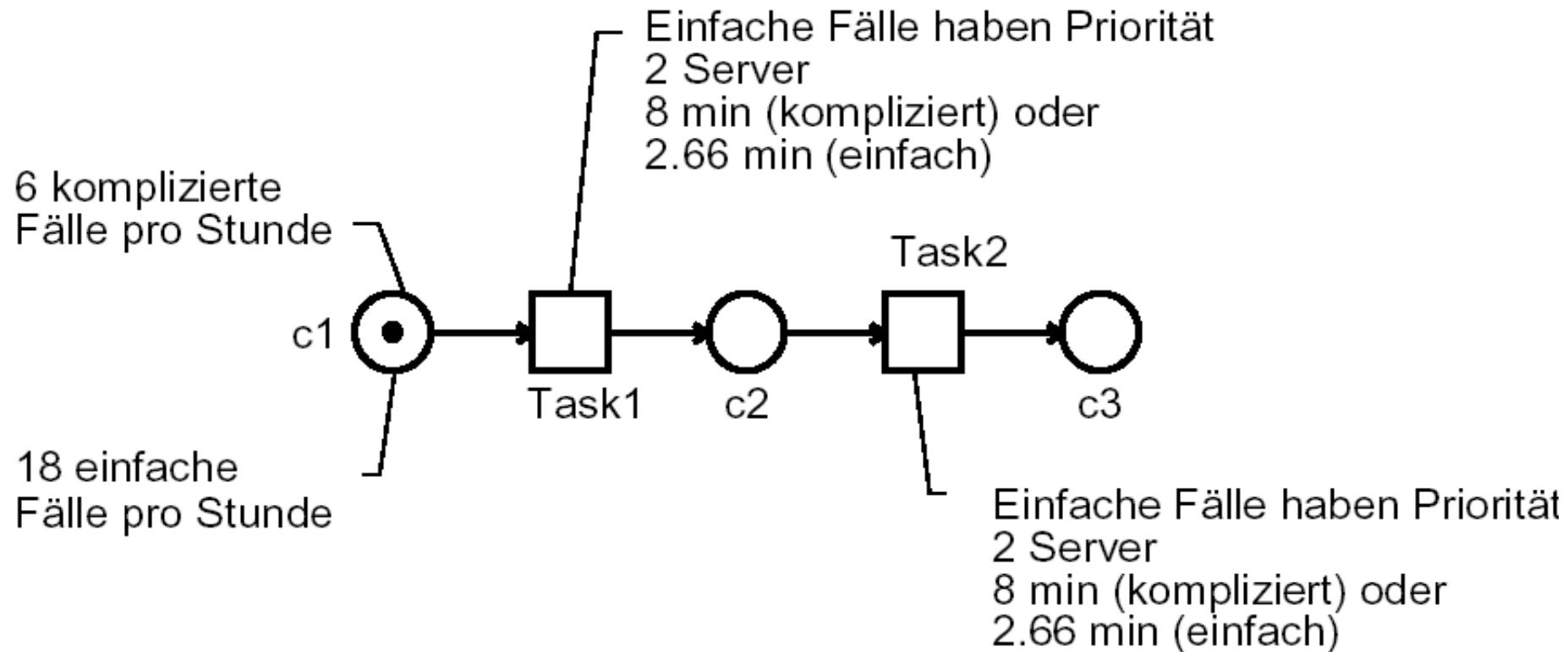
Durchlaufzeit : 31.1 min

Servicezeit: 8.0 min

Wartezeit: 23.1 min

7. Performance-Analyse / Simulation

4. Szenario 6: Priorität



Durchlaufzeit : 14.0 min

Servicezeit: 8.0 min

Wartezeit: 6.0 min

7. Performance-Analyse / Simulation

4. Übersicht der Ergebnisse

Szenario	Beschreibung	durchschn. Durchlaufzeit	durchschn. Servicezeit	durchschn. Wartezeit
Situation 1	sequentiell	22.2	8.0	14.2
Situation 2	parallel	15.0	4.0	11.0
Situation 3	Komposition	9.5	7.0	2.5
Situation 4	Flexible Ress.	14.0	8.0	6.0
Situation 5	Triage	31.1	8.0	23.1
Situation 6	Priorität	14.0	8.0	6.0

Fragebogen GPM WS 2018/19

1. Grundlagen

- a) Welche Aufgaben soll GPM erfüllen. **Einführung Folie 8**
- b) Was ist unter dem strategischen Ansatz/ Software Toolset von ECM zu verstehen? **Einführung Folie 28**
- c) Wozu dient das Workflow/ Business Process Management (BPM)? **Einführung Folie 33**
- d) Welche Geschäftsprozesse kennen Sie? **Einführung Folie 9**
- e) Welche positiven Aspekte umfasst die Realisierung vorgangsorientierter Anwendungssysteme mittels prozessorientiertem Workflow-Management-System? **Einführung Folie 24**
- f) Welche Einschränkungen gibt es heute noch? **Einführung Folie 24**

2. Prozesse, Prozess-Management, Modelle

- a) Was versteht man unter einem Geschäftsprozess? **Prozesse_und_Management_1 Folie 7**
- b) Welche Elemente umfasst ein Prozess typischerweise? **Prozesse_und_Management_1 Folie 7**
- c) Wie lassen sich Geschäftsprozesse klassifizieren? **Prozesse_und_Management_1 Folie 9ff**
- d) Welche Art von Aufgaben lassen sich gut automatisieren? **Prozesse_und_Management_1 Folie 12**
- e) Was verstehen Sie unter einem Prozessmodell? **Prozesse_und_Management_1 Folie 16**
- f) Was versteht man unter einer Ressourcenklasse? **Prozesse_und_Management_1 Folie 24**
- g) Erklären Sie vier Möglichkeiten für das Routing von Fällen nach Aalst an. **Prozesse_und_Management_1 Folie 31**
- h) Welche vier Kriterien zur Bewertung des Prozessdesigns gibt es? Welches Problem tritt bei der Erreichung auf? **Prozesse_und_Management_1 Folie 41**
- i) Beschreibung der Kriterien zum Prozessdesign. **Prozesse_und_Management_1 Folie 41ff**

3. Prozess Verbesserung und-Modellierung

- a) Welche Elemente gehören zu einer EPK, einem Petrinetz, BPMN 2.0?
- b) Was versteht man unter einem Bedingungs-/ Ereignis-Netz?
- c) Welche häufig verwendeten Sequenzmuster bei Petrinetzen kennen Sie?
- d) Unter welchen Bedingungen schaltet eine Transition?
- e) Was versteht man unter einem Konflikt, Synchronisation, Split in einem Petrinetz?
- f) Zeichnen Sie ein Petrinetz, in dem ein Deadlock entstehen kann.
- g) Zeichnen Sie ein Petrinetz, in dem kein Deadlock vorkommt.
- h) Welche Aussagen sind wahr?
 - Jedes lebendige Petrinetz mit mindestens einer Transition ist verklemmungsfrei.
 - Es gibt kein Petrinetz, welches terminiert und verklemmungsfrei ist.
 - Ein Petrinetz ist genau dann verklemmungsfrei, wenn der dazugehörige Markierungsgraph keinen Knoten ohne Nachfolger besitzt.
- i) Was versteht man unter einem stellen-/ transitionsberandetem Petrinetz?
- j) Welche Bedeutung hat das Symbol „[m]>“?
- k) Nennen Sie typische Bestandteile von High-Level-Petrinetzen
- l) Gegeben ist ein das Petrinetz N: Zeichnen Sie das Petrinetz, vergessen Sie die Startmarkierung nicht.

S={P0,P1,P2,P3,P4,P5}

T={T0,T1,T2,T3,T4,T5}

M0 = (1,0,0,0,0,1)

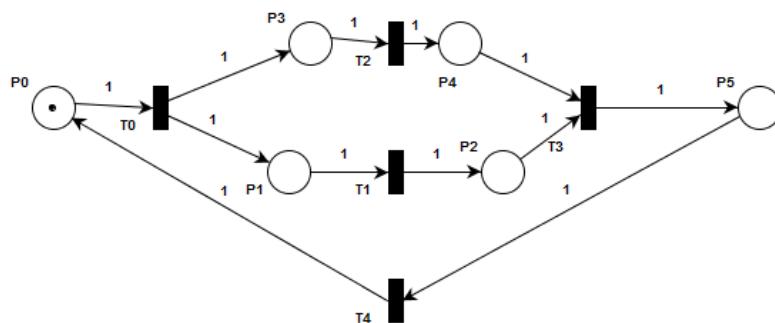
F={(P0,T0),(T0,P1),(T0,P3),(P3,T2),(T2,P4),(P4,T3),(P1,T1),(T1,P2),(T2,P3),(T3,P5),(P5,T4),(T4,P0)}

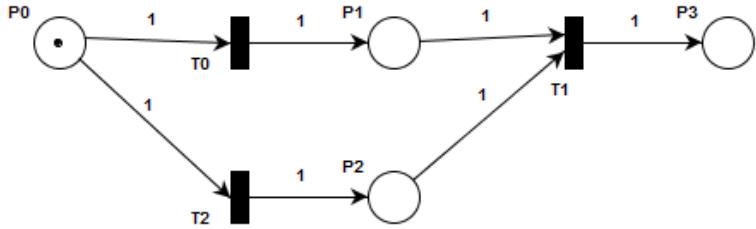
- m) Was versteht man unter BPR? **(Verbesserung und-Modellierung Folie 5)**
- n) Was sind Symptome für einen Verbesserungsbedarf? **Verbesserung und-Modellierung Folie 6**
- o) Was versteht man unter CPI? **Verbesserung und-Modellierung Folie 8ff**
- p) Vergleichen Sie CPI und BPR. **Verbesserung und-Modellierung Folie 14ff**

- q) Was versteht man unter einem Modell? **Verbesserung und-Modellierung Folie 20**
- r) Welche Ansätze zur Modellierung gibt es? **Verbesserung und-Modellierung Folie 34**

4. Prozess-Analyse

- Was versteht man unter Free-Choice?
- Welche Struktureigenschaften hat ein korrektes Workflow-Prozess-Modell?
- Wann ist ein Petrinetz „save“, wann beschränkt?
- Was versteht man unter „state space explosion“?
- Welche Konnektoren gibt es bei EPKs?
- Was ist ein Event in einer EPK?
- Was ist eine Funktion in einer EPK?
- Wie dürfen Event und Funktion (bei EPK) mit den Konnektoren verbunden sein?
- Wann ist ein Petrinetz lebendig/ tot?
- Wann ist eine Transition in einem Petrinetz lebendig?
- Wann ist eine Markierung in einem Petrinetz lebendig?
- Zeichnen Sie bitte für die folgenden Petrinetze den Markierungsgraphen!





- m) Unter welchen Bedingungen ist ein Work-Flow-Netz sound?
- n) Geben Sie die Inzidenzmatrix für die beiden Petrinetze aus I an.
- o) Wie wird die S-Invariante berechnet?
- p) Wie wird die T-Invariante berechnet?
- q) Gegeben ist folgende S-Invariante: (1,2,1,1). Welche Aussage können Sie über die Beschränktheit des Petrinetzes treffen?
- r) Was versteht man unter Deadlock, Livelock, Rerversibilität und Terminiertheit bei Petrinetzen?
- s) Zeichnen Sie ein Petrinetz, für welches kein sinnvoller Markierungsgraph gezeichnet werden kann, sondern ein Überdeckungsgraph sinnvoll ist.

5. BPMN 2.0

- a) Wozu dienen Gateways in BPMN 2.0?
- b) Was versteht man unter BPMN? Was sind die Ziele? Wo Zielkonflikte? (**BPMN Folie 7ff**)
- c) Was versteht man unter einem Ereignis bzw. Event in BPMN 2.0? (**BPMN Folie 19**)
- d) Welche Elemente gibt es in BPMN, in welche Kategorien lassen sie sich einteilen? (**BPMN Folie 13ff**)
- e) Was sind Ablaufelemente in BPMN 2.0? Zeichnen und benennen Sie drei Stück. (**BPMN Folie 17**)
- f) Welche Gateways kennen Sie? Beschreiben Sie diese. (**BPMN Folie 20ff und Praktikum**)
- g) Was versteht man unter einem intermediate Catching Event?
- h) Was ist ein Boundary Event?
- i) Beschreiben Sie Timer-Events. An welchen Stellen in einem BPMN 2.0-Diagramm können Sie eingesetzt werden? Wie wirken sie dort?
- j) Was sind Pools und (Swim-)lanes in BPMN 2.0? (**BPMN Folie 22**)

- k) Was ist der Unterschied zwischen einer User-Task und einer Business Rule Task bei BPMN 2.0? Wie sehen die Symbole aus?
(GPM_Praktikum_4_BPMN20)
- l) Wie unterscheiden sich Signal Events von Message Events bei BPMN 2.0?
(GPM_Praktikum_4_BPMN20)
- m) Nennen Sie neun Best Practices für die Modellierung mit BPMN 2.0.
(GPM_Praktikum_4_BPMN20)