

# ***Software-Engineering***

Sebastian Iwanowski  
FH Wedel

**Kapitel 4: Systemanalyse**  
**Teil 1: Prozessorientierte Sicht**

# Überblick: Systemanalyse

## Verschiedene Analysemethoden unterscheiden sich in den Sichten:

- Funktional-hierarchische Sicht
- Datenflussorientierte Sicht
- Algorithmische Sicht
- Ereignisorientierte Sicht
- Zustandsorientierte Sicht
- Regelbasierte Sicht
- Daten-Beziehungs-Sicht
- Objektorientierte Sicht
- Geschäftsprozessorientierte Sicht
- ...

## 2 grobe Sichteinteilungen:

- Prozessorientierte Sicht
- Datenorientierte Sicht

# Überblick: Systemanalyse

## Gliederung für die Vorlesung Software-Engineering:

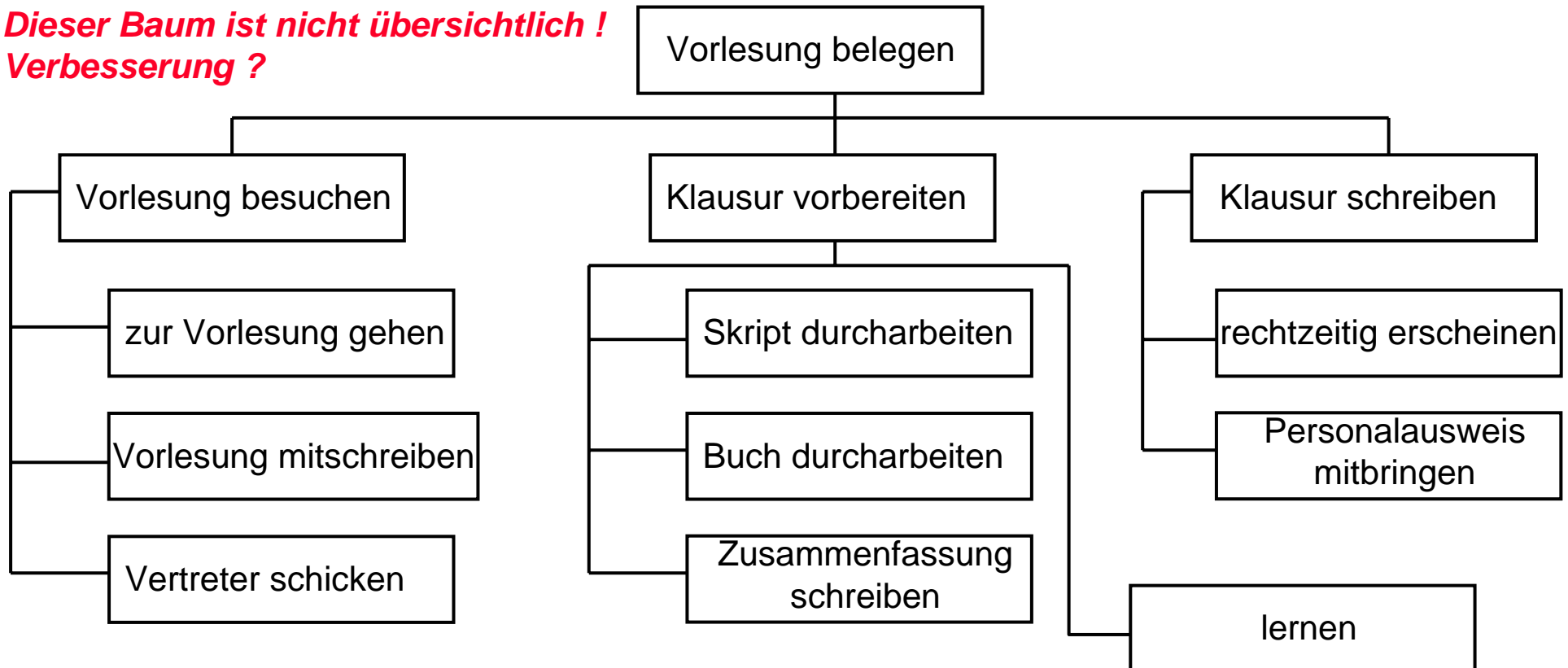
1. Prozessorientierte Sicht
2. Datenorientierte Sicht
3. UML in einem CASE-Tool
4. ARIS

# Prozessorientierte Sicht: Hierarchische Zerlegung

## Hierarchische Zerlegung:

- Zerlege einen Prozess in Teilprozesse
  - Graphische Darstellung als Funktionsbaum

*Dieser Baum ist nicht übersichtlich !  
Verbesserung ?*



# Darstellungsart Funktionsbaum

## Elemente des Funktionsbaums:

- **Rechteck:** Prozess / Funktion
- **Kante:** Hierarchiebeziehung zwischen Funktionen

**Kante nach unten** führt immer zur hierarchisch tieferen Funktion

**Kante nach oben** führt immer zur hierarchisch höheren Funktion

**Kante zur Seite** muss an vertikale Kante angeschlossen sein:

Wenn diese vertikale Kante weiter nach oben führt, dann ist die Funktion an der horizontalen Kante eine hierarchisch tieferstehende Funktion.

Wenn die vertikale Kante nur nach unten führt, dann ist die Funktion an der horizontalen Kante eine hierarchisch höherstehende Funktion.

***Hier werden nur Bausteine aufgezählt,  
keine Beziehungen zwischen ihnen !***

# Prozessorientierte Sicht: Modulare Zerlegung

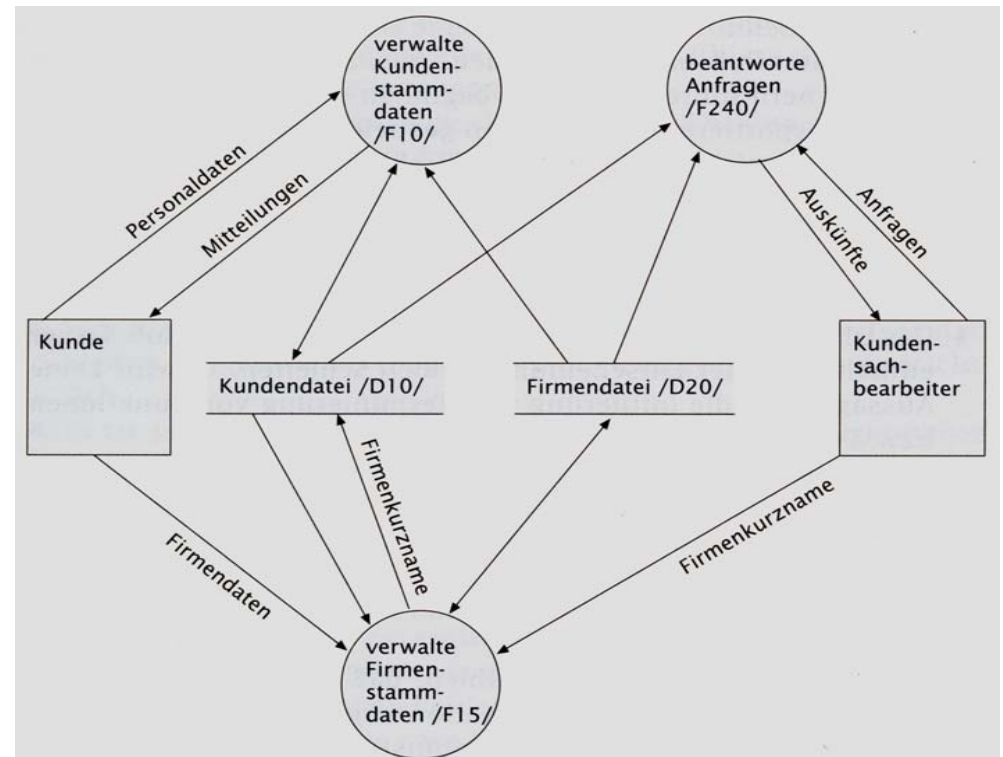
## Modulare Zerlegung:

- Stelle die **Beziehung** verschiedener Prozesse untereinander dar
  - Graphische Darstellung als Datenflussdiagramm
  
- Stelle den **Kontrollfluss** eines Prozesses dar:
  - Sequenzen
  - Verzweigungen
  - Schleifen
  - Graphische Darstellung als Struktogramm oder Programmablaufplan
  - Spezielle Darstellungsform Entscheidungsbaum /-tabelle

# Darstellungsart Datenflussdiagramm (DFD)

## Elemente des Datenflussdiagramms:

- Kreis: Prozess / Funktion
- Rechteck: (Daten)-Schnittstelle
- Offenes Rechteck: Datenspeicher
- Pfeil: Datenflussrichtung



[aus Balzert 1, Kap. 2.7]

# Darstellungsart Datenflussdiagramm (DFD)

## Regeln für ein Datenflussdiagramm:

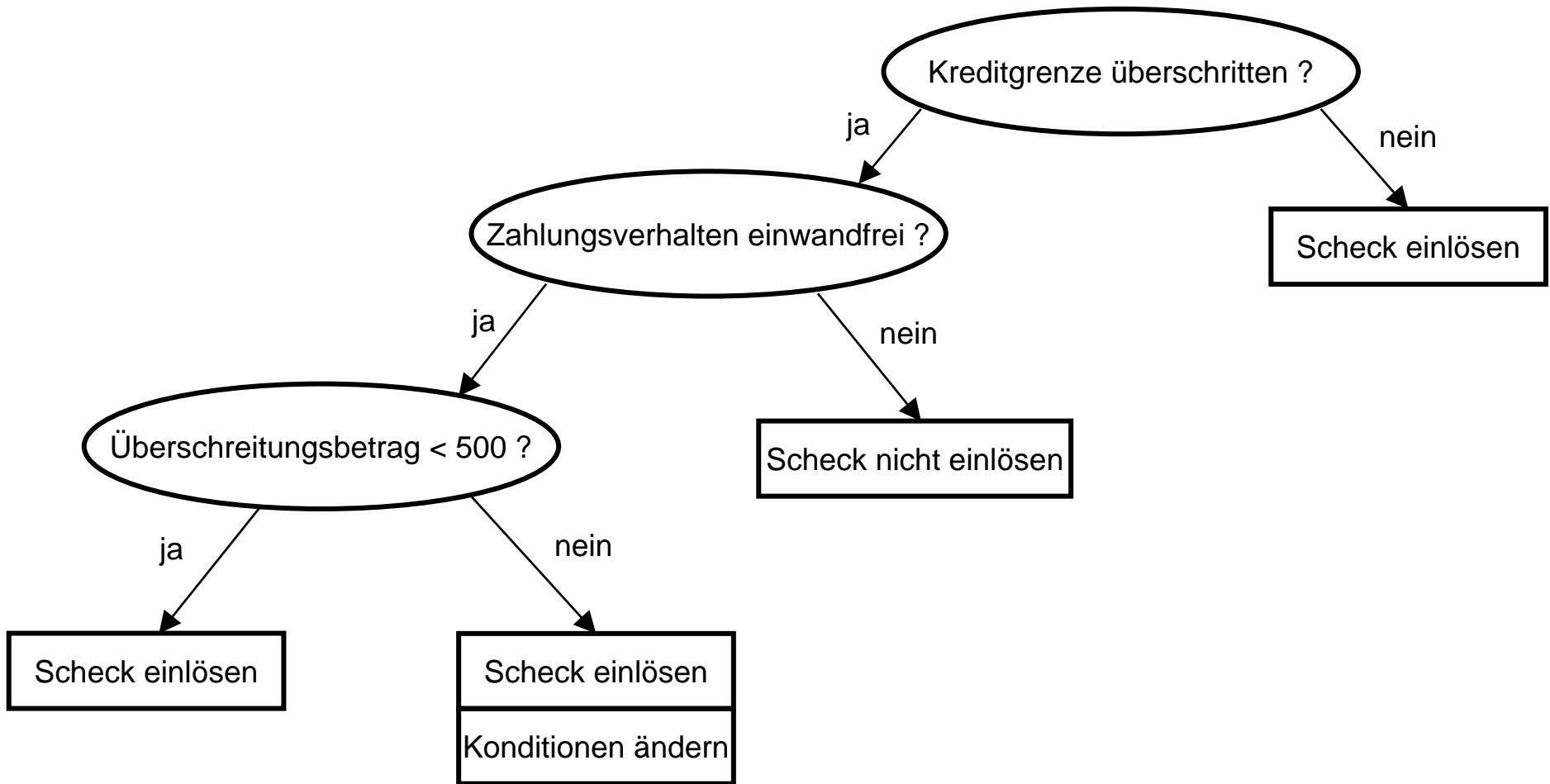
- Jedes Diagramm hat mindestens einen Prozess.
- Keine Datenflüsse zwischen Schnittstellen.
- Kein Prozess oder Schnittstelle ohne Datenfluss.
- Jeder Datenspeicher wird geschrieben und gelesen.
- Jeder Prozess ist eindeutig nummeriert.
- Alle Elemente haben eine eindeutige Bezeichnung.

***Hier werden nur Beziehungen zwischen den Bausteinen angegeben, kein Ablaufplan des Kontrollflusses !***



# Darstellungsart Entscheidungstabelle /-baum

## Entscheidungsbaum

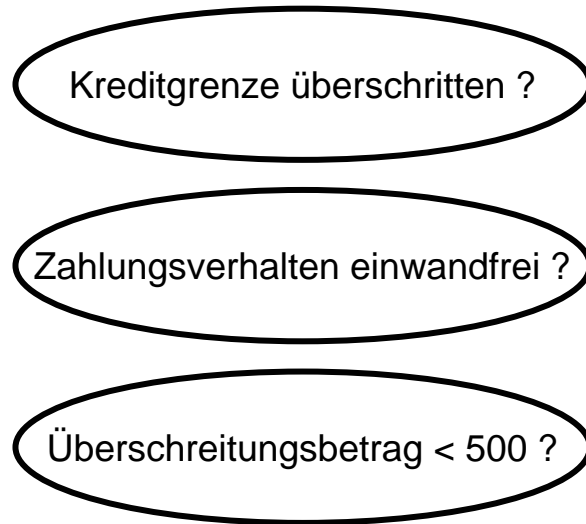


[nach Balzert 1, Kap. 2.14]

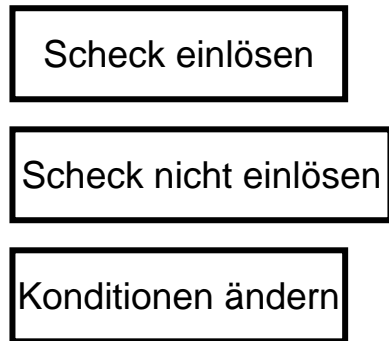
# Darstellungsart Entscheidungstabelle /-baum

## Entscheidungstabelle

**Bedingungen**



**Aktionen**



alle Möglichkeiten

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| j | j | j | j | n | n | n | n |
| j | j | n | n | j | j | n | n |
| j | n | j | n | j | n | j | n |
| x | x |   |   | x |   | x |   |
|   |   | x | x |   |   |   |   |
| x |   |   |   |   |   |   |   |

[nach Balzert 1, Kap. 2.14]

# Darstellungsart Entscheidungstabelle /-baum

## Entscheidungstabelle (horizontale Form)

| Kreditgrenze überschritten? | Zahlungsverhalten einwandfrei? | Überschreibungsbetrag < 500,-? | Scheck einlösen | Scheck nicht einlösen | neue Konditionen vorlegen | unlogisch |
|-----------------------------|--------------------------------|--------------------------------|-----------------|-----------------------|---------------------------|-----------|
| J                           | J                              | J                              | X               |                       |                           |           |
|                             |                                | N                              | X               |                       | X                         |           |
|                             | N                              | J                              |                 | X                     |                           |           |
|                             |                                | N                              |                 | X                     |                           |           |
| N                           | J                              | J                              | X               |                       |                           |           |
|                             |                                | N                              |                 |                       |                           | X         |
|                             | N                              | J                              | X               |                       |                           |           |
|                             |                                | N                              |                 |                       |                           | X         |

[aus Balzert 1, Kap. 2.14]

# Zusammenfassung: Strukturierte Analyse

Verwendung und Verfeinerung der bisher vorgestellten Methoden in:

**Strukturierte Analyse (SA)** (Tom deMarco, 1978)

**Bestandteile:**

Datenflussdiagramme (hierarchisch verschachtelt)

*Data Dictionaries (datenorientierter Anteil der SA)*

Darstellung von Kontrollflüssen (nicht genormt)

Verwendung von Entscheidungsbäumen und -tabellen

**Erweiterung der Strukturierten Analyse:**

**Real Time Analysis (SA/RT)** (Derek Hatley, 1987)

Erweiterte Beschreibungsmöglichkeiten aus ereignisorientierter Sicht

# EPK: Ereignisorientierte Prozessketten



- Ereignisse lösen Prozesse aus.
- Prozesse bedingen Ereignisse.
- Die Kontrollflüsse dürfen sich verzweigen.
- In jedem Kontrollfluss wechseln sich Ereignisse und Prozesse ab.
- Jeder Kontrollfluss beginnt und endet mit einem Ereignis.

# EPK: Ereignisorientierte Prozessketten

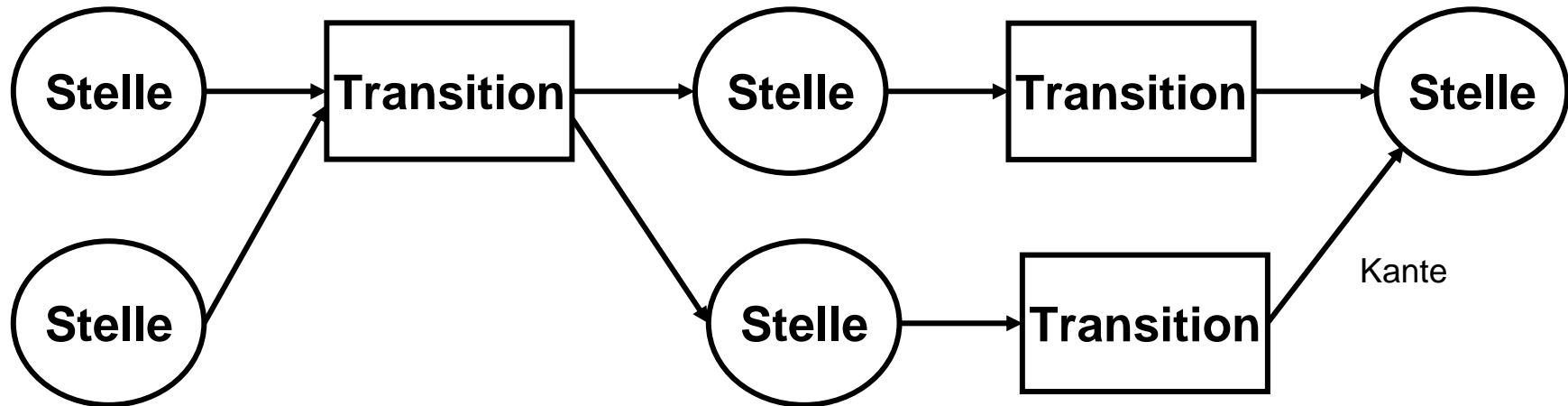
## Verzweigungen von Kontrollflüssen in EPKs

| Verknüpfungsart \ Verknüpfungsoperatoren |                            | exklusives oder | und | oder (don't care) |
|--|----------------------------|-----------------|-----|-------------------|
|  |                            |                 |     |                   |
| Ereignisverknüpfung                      | Auslösende Ereignisse (AE) |                 |     |                   |
|  | Erzeugte Ereignisse (EE)   |                 |     |                   |
| Funktionsverknüpfung                     | Auslösende Ereignisse (AE) |                 |     |                   |
|  | Erzeugte Ereignisse (EE)   |                 |     |                   |

**Legende:**

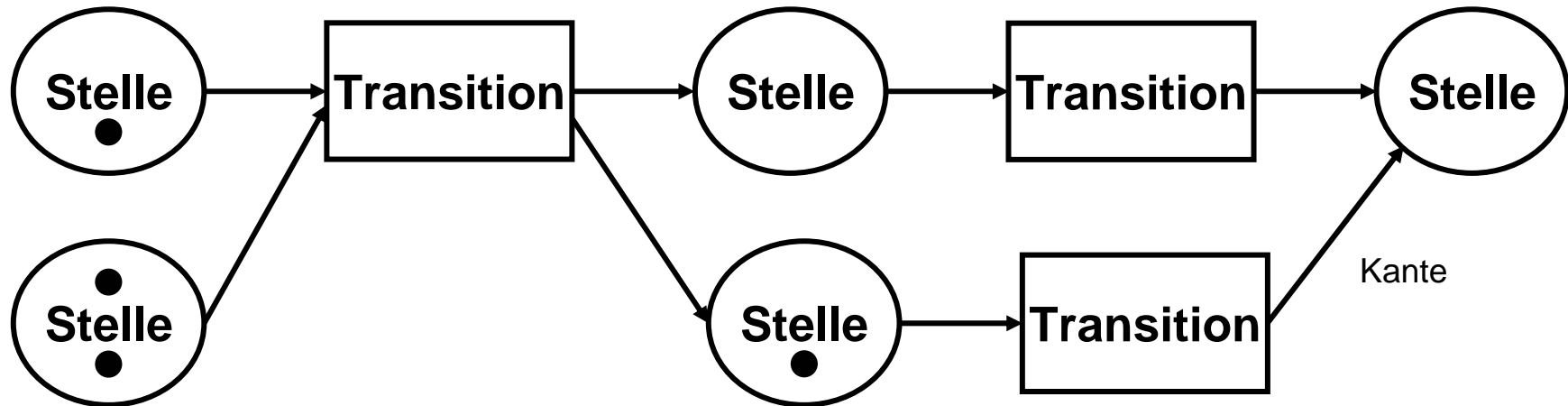
- = und
- = exklusives oder
- = oder (don't care)
- = nicht erlaubt
- E = Ereignis
- F = Funktion

# Petri-Netze



- Stellen sind die Inputs und Outputs von Transitionen.
- Stellen dürfen mit mehreren Transitionen verbunden sein (auf beiden Seiten).
- Transitionen dürfen mit mehreren Stellen verbunden sein (auf beiden Seiten).
- Die Kanten sind grundsätzlich gerichtet.
- Jeder Kontrollfluss beginnt und endet mit einer Stelle. (nicht immer gefordert)

# Petri-Netze



Eine Stelle kann mit Marken (Tokens) belegt werden.

Für jede Transition gibt es eine Schaltregel, die den Übergang der Marken von Eingabe- zu Ausgabestellen beschreiben

- Eine Transition kann schalten ("feuern"), wenn jede Eingabestelle mindestens eine Marke enthält
- Schaltet eine Transition, dann wird aus jeder Eingabestelle eine Marke entfernt und zu jeder Ausgabestelle eine Marke hinzugefügt



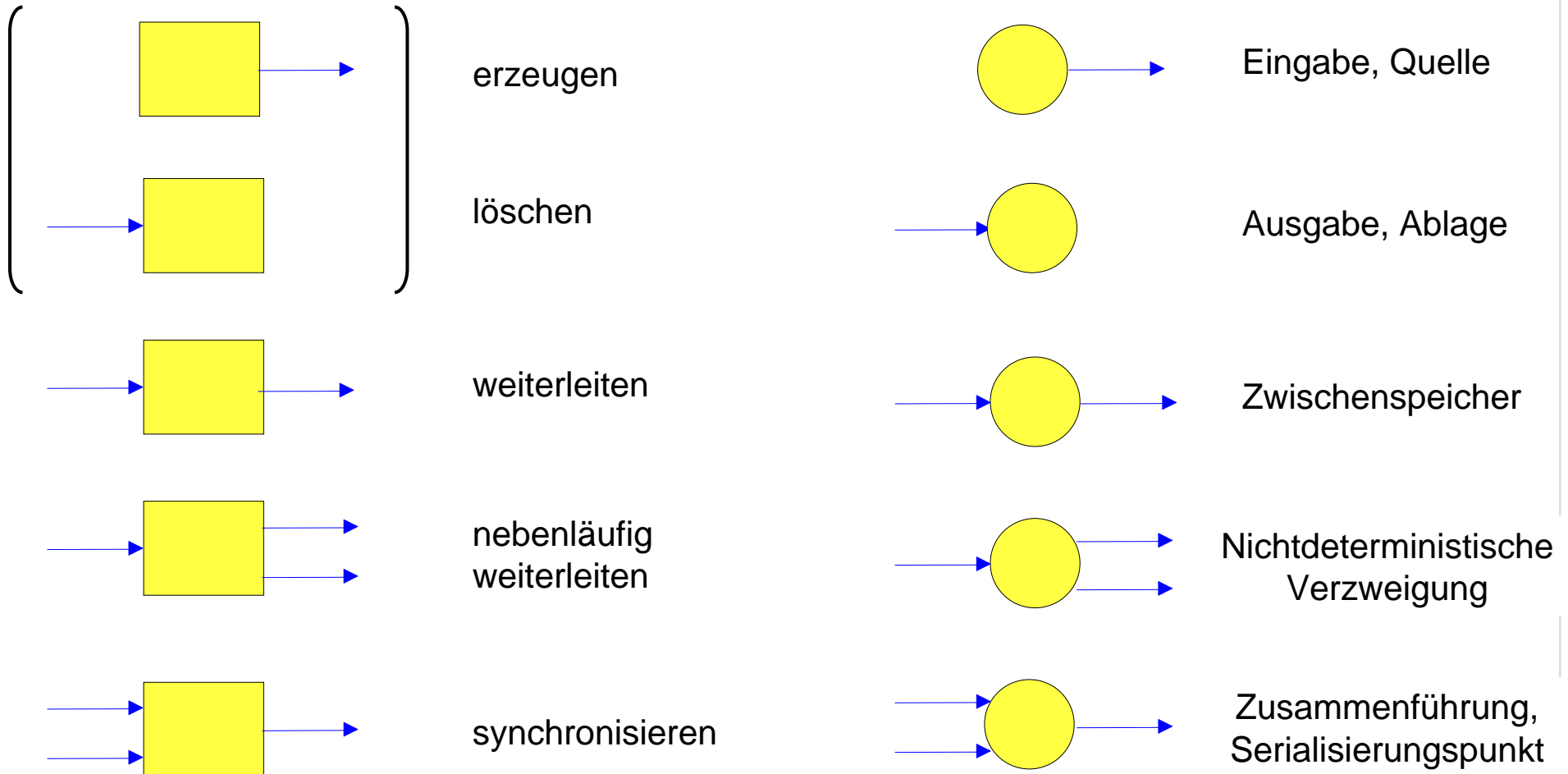
# Petri-Netze

## Historie und Bedeutung

- erfunden von **Carl Adam Petri** Anfang der sechziger Jahre (Dissertation plus Weiterentwicklung)
- Modellierungsmethodik für dynamische Systeme (Zustandsübergänge)
- besonders geeignet für nebenläufige und nicht-deterministische Vorgänge (vor allem in **verteilten Systemen**)
- verallgemeinert die Konzepte von EPKs
- vielfältig theoretisch analysiert
- dient zur theoretischen Analyse auch komplexer Vorgänge
- viele Simulationswerkzeuge erhältlich

# Petri-Netze

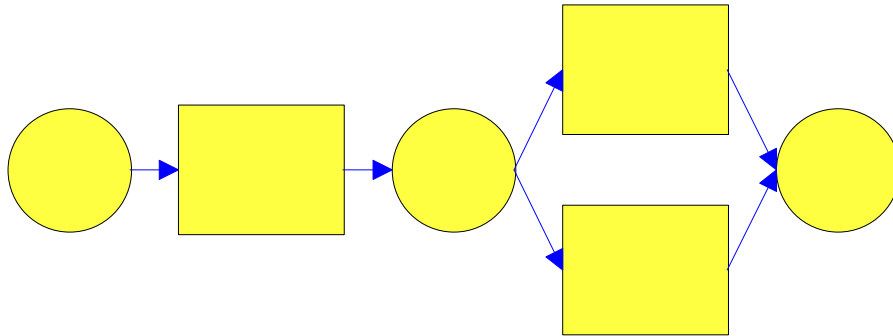
## Bausteine für Petri-Netze



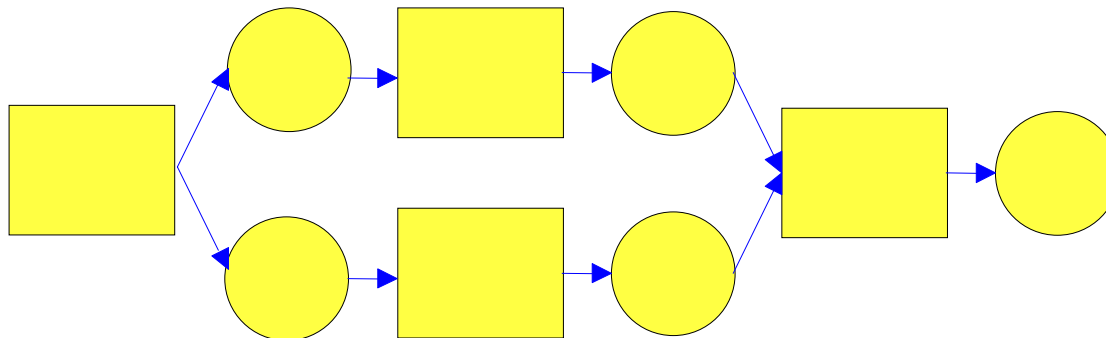
**Mit diesen Bausteinen kann man alle Netze zusammenbauen.**

# Petri-Netze

## nichtdeterministischer Verlauf



## nebenläufiger Verlauf mit Synchronisation

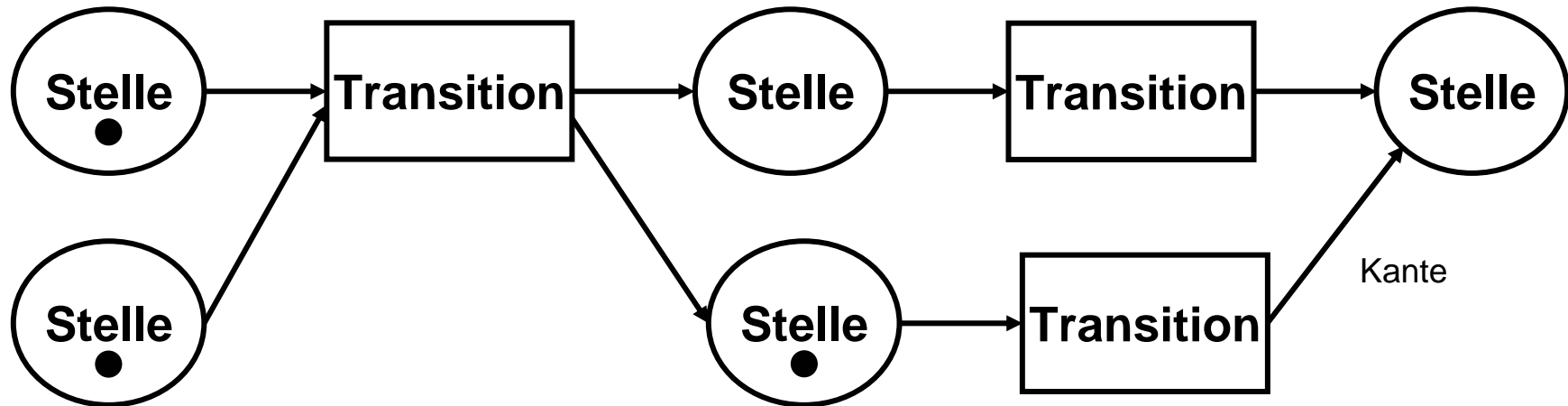


# Petri-Netze

## Typen von Petri-Netzen

- **Bedingungs/Ereignis-Netze**
- **Stellen/Transitions-Netze**
- **Prädikat/Transitions-Netze**

# Bedingungs/Ereignis-Netze

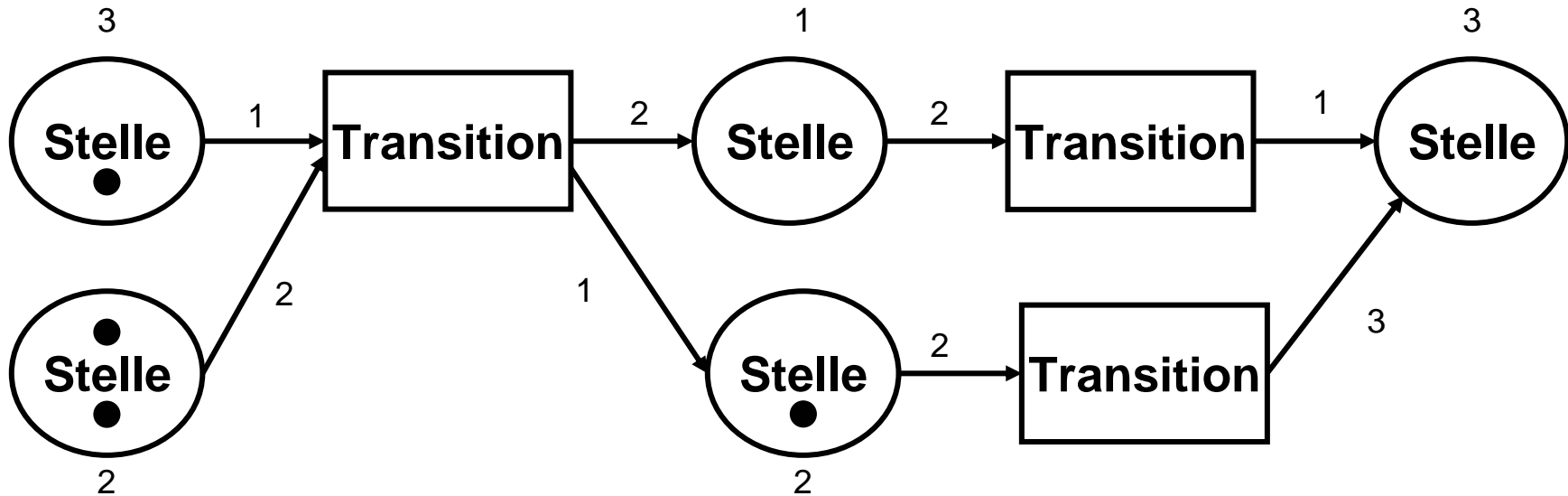


Marken realisieren den Datentyp Boolean,  
d.h. jede Stelle kann genau eine oder keine Marke enthalten

Folgerung für die Schaltregel:

- Eine Transition kann schalten, wenn jede ihrer Eingabestellen eine Marke enthält und jede ihrer Ausgabestellen leer ist

# Stellen/Transitions-Netze (P/T Nets)



Das „normale“ Petri-Netz:

Zusätzliche Restriktionen für:

- Stellenkapazitäten: maximal zulässige Anzahl von Marken
- Kantengewichte: Anzahl der Marken, die beim Schalten transportiert werden

Folgerung für die Schaltregel:

- Eine Transition kann schalten, wenn jede ihrer Eingabestellen mindestens so viele Marken enthält wie das Gewicht der dahinterliegenden Eingabekante und jede ihrer Ausgabestellen höchstens so viele Marken enthält wie ihre Kapazität minus das Gewicht der davorliegenden Ausgabekante

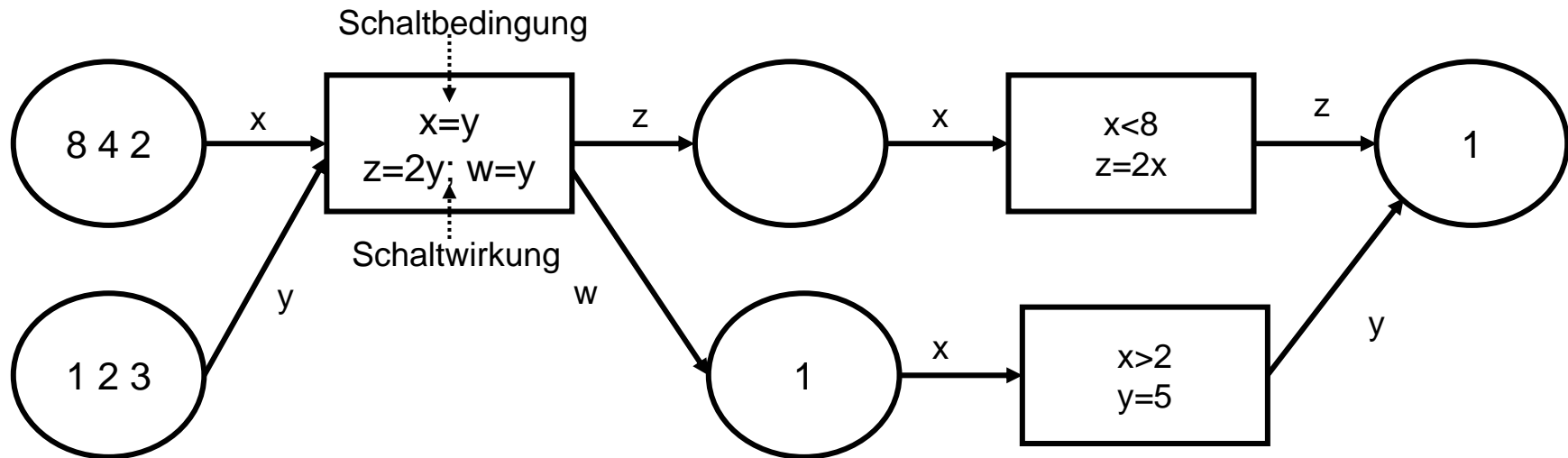
# Stellen/Transitions-Netze (P/T Nets)

## Modellierungsaufgabe: Hotelportiers

Ein Hotel hat drei fest angestellte Portiers.

Es muss mindestens einer arbeiten, aus Kostengründen dürfen niemals alle drei gleichzeitig tätig sein.

# Prädikat/Transitions-Netze



Marken realisieren einen beliebigen Datentyp, z.B. Integer.

In den Transitionen werden gesondert eine Schaltbedingung und eine Schaltwirkung angegeben.

Folgerung für die Schaltregel:

- Eine Transition kann schalten, wenn jede ihre Schaltbedingung mit entsprechenden Marken aus den Eingabestellen versorgt werden kann.
- Die Marken, die sich aus der Schaltwirkung ergeben, werden **zusätzlich** in den Ausgabestellen eingefügt.



# Prädikat/Transitions-Netze

## Modellierungsaufgabe: Fertigungsprozess

Ein Unternehmen stellt zwei Endprodukte  $\{E1, E2\}$  her. Hierzu benötigt es zwei Basisstoffe  $\{B1, B2\}$ . Auf Maschine 1 werden die beiden Basisstoffe  $\{B1, B2\}$  zum Zwischenprodukt Z1 weiterverarbeitet.

Die Maschine 2 stellt aus dem Basisstoff B2 das Zwischenprodukt Z2 her. Z1 wird auf der Maschine 3 zu Zwischenprodukt Z4 und mit dem Zwischenprodukt Z1 auf der Maschine 4 zu Endprodukt 1 zusammengesetzt. Z2 wird auf der Maschine 5 zu Endprodukt 2 verarbeitet.

# Petri-Netze

## Weitere Funktionalitäten / Erweiterungen:

- Hierarchische Petri-Netze
- Konfliktlösungsmechanismen

## Wert für die Systemanalyse:

- erlaubt die Analyse technisch schwieriger Probleme auf eine sehr systematische Weise
- führt zu mathematisch beweisbaren Aussagen über das Systemverhalten
- Ideen auch anderswo verwendbar (z.B. in UML2)
- wird im “Normalfall” nicht direkt eingesetzt

# BPMN: Business Process Modelling Notation

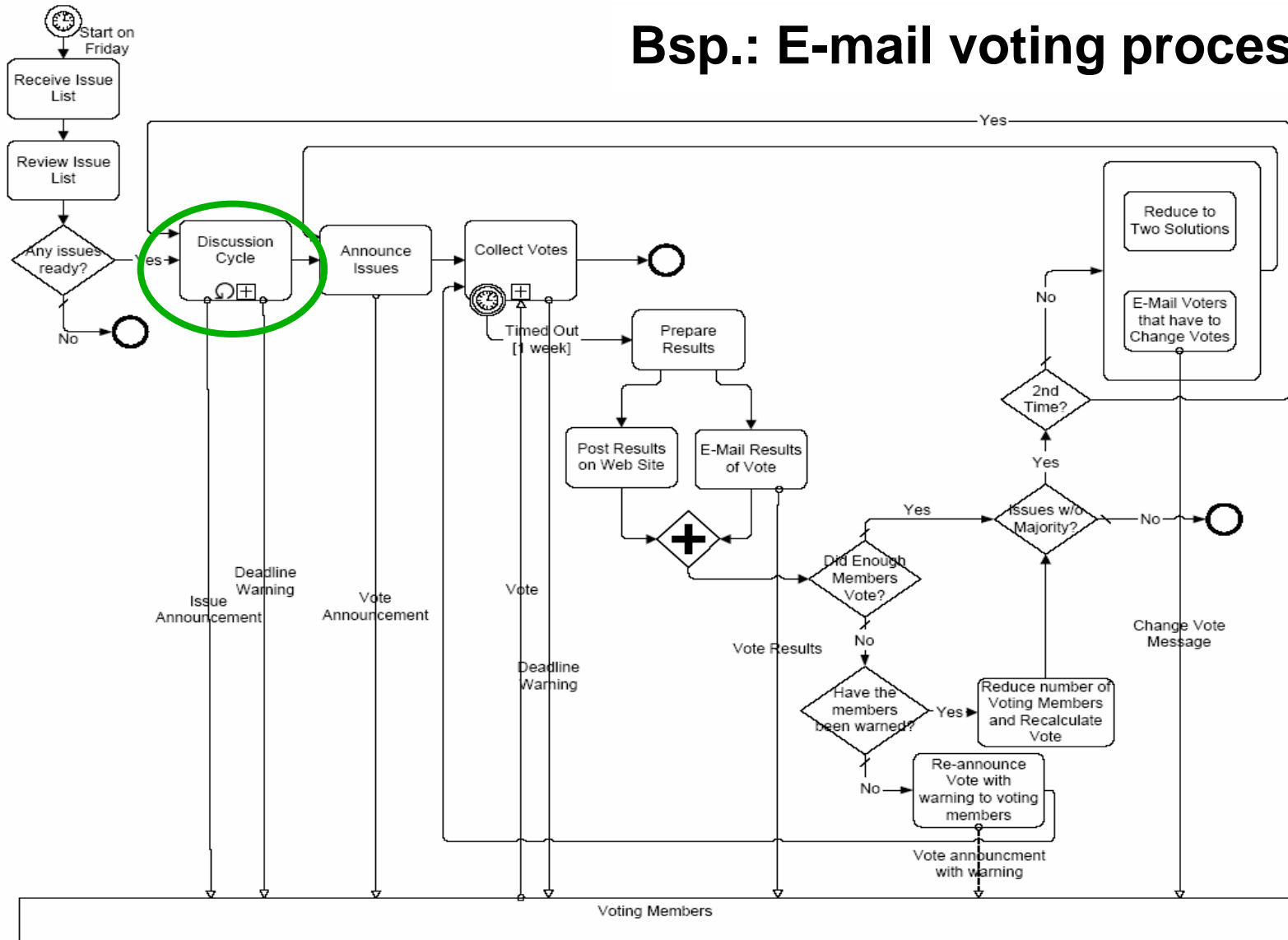
## der Nachfolger von EPKs ?

### Zusätzliche Funktionalitäten / besondere Eigenschaften:

- Modellierung genauer zeitlicher Bedingungen
- verschiedene Ereignistypen (Nachricht, Fehler, etc.)
- verschiedene Abstraktionsstufen
- explizite Modellierung verschiedener Organisationseinheiten
- sehr verständliches graphisches Layout

# BPMN: Business Process Modelling Notation

## Bsp.: E-mail voting process



# BPMN: Business Process Modelling Notation

## Bsp.: Discussion Cycle

