



10	Ereignisgesteuerte Prozessketten (EPK)	
10.1	Einführung	2
10.2	Mögliche Anwendungsgebiete	3
10.3	Grundobjekte: EPK-Modellierung	4
10.4	Grundregeln der EPK-Modellierung	12
10.5	ARIS (Architektur integrierter Informationssysteme)	21
10.6	Erweiterte EPK und ARIS	22
10.7	EPK - Petrinetze	27
10.8	Bewertung EPK im Hinblick auf Petrinetze	35



AIFB

27.01.2003

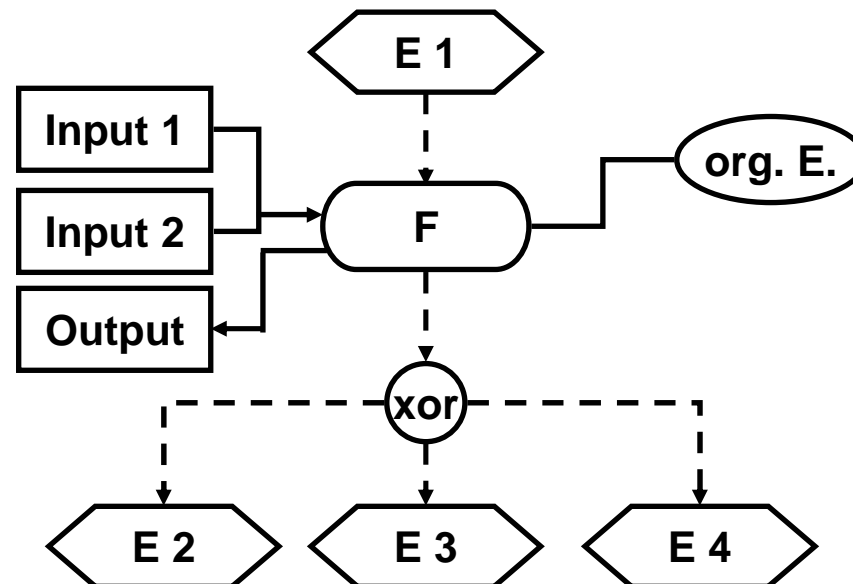
EPK





10.1 Einführung

- semiformale, graphische Modellierungssprache
- in 1992 entwickelt von Prof. Scheer (IDS Prof. Scheer GmbH) und Mitarbeitern
- hoher Verbreitungsgrad in Deutschland: ARIS-Toolset bzw. SAP R/3-Analyzer
- EPKs beschreiben Prozesse (zusammenhängende Aktivitäten / Ablaufreihenfolgen)





10.2 Mögliche Anwendungsgebiete

(nach Loos, Allweyer: 1998)

- Business process re-engineering (BPR)
- Definition und Kontrolle von Workflows
- Konfiguration von Standard-Software (Standardsoftwareanbieter wie SAP verwenden EPK für Prozeßdokumentation)
- Softwareentwicklung
- Simulation
- Activity-based costing (ABC)
- Prozeßdokumentation nach ISO900x





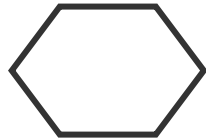
10.3 Grundobjekte: EPK-Modellierung

(1|8)

Eine 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)



10.3 Grundobjekte: EPK-Modellierung

(2|8)

In ihrem **Grundprinzip** besteht die EPK aus

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

Ereignisse

Ereignisse sind

- Auslöser von Funktionen und
- deren Ergebnis.

Ereignisse repräsentieren einen eingetretenen **betriebswirtschaftlichen Zustand**.



AIFB

27.01.2003

EPK



10.3 Grundobjekte: EPK-Modellierung

(3|8)

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*



10.3 Grundobjekte: EPK-Modellierung

(4|8)

Funktionen

Funktionen transformieren Input- in Outputdaten
(siehe 10.6 Erweiterte EPK)

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 | ...



10.3 Grundobjekte: EPK-Modellierung

(5|8)

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**.



10.3 Grundobjekte: EPK-Modellierung

(6|8)

Verknüpfungsoperatoren



disjunktive Verknüpfung XOR; „entweder-oder“- Verknüpfung:

- *die Gesamtaussage ist wahr, wenn genau eine Aussage wahr ist*

konjunktive Verknüpfung AND; „und“-Verknüpfung:

- *die Gesamtaussage ist wahr, wenn beide Aussagen gleichzeitig wahr sind*

adjunktive Verknüpfung OR; „und/oder“-Verknüpfung:

- *die Gesamtaussage ist wahr, wenn mindestens eine Aussage wahr ist*

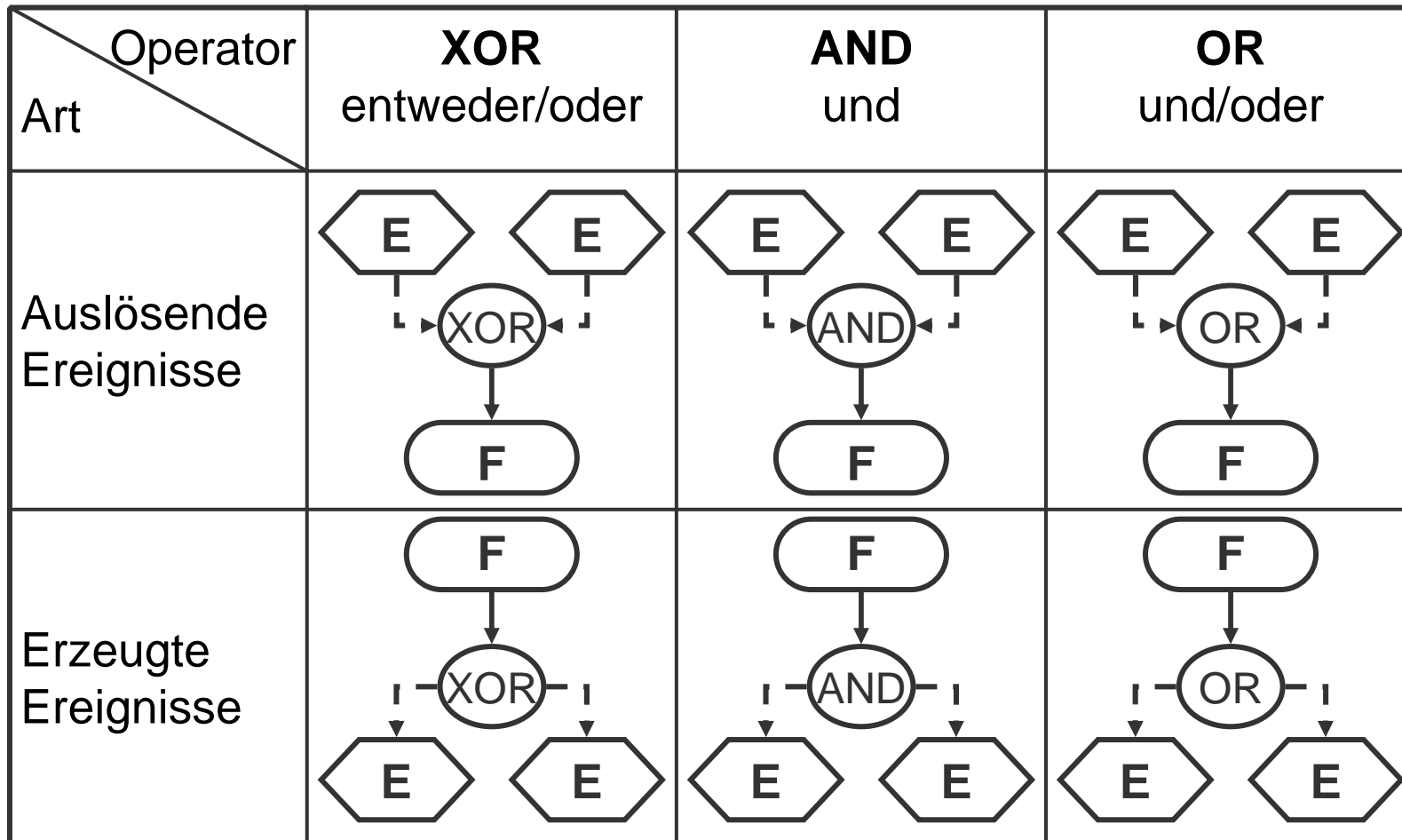


10.3 Grundobjekte: EPK-Modellierung

(7|8)

Ereignisverknüpfung (syntaktische Möglichkeiten):

- Mehrere Ereignisse werden mit einer Funktion verknüpft



10.3 Grundobjekte: EPK-Modellierung

(8|8)

Funktionsverknüpfung (syntaktische Möglichkeiten):

- Mehrere Funktionen werden mit einem Ereignis verknüpft

Operator \ Art	XOR entweder/oder	AND und	OR und/oder
Auslösendes Ereignis [Vgl. aber 10.4 (4 6)]	<pre> graph TD E{{E}} --> XOR((XOR)) XOR --> F1(F) XOR --> F2(F) </pre>	<pre> graph TD E{{E}} --> AND((AND)) AND --> F1(F) AND --> F2(F) </pre>	<pre> graph TD E{{E}} --> OR((OR)) OR --> F1(F) OR --> F2(F) </pre>
Erzeugtes Ereignis	<pre> graph TD F1(F) --> XOR((XOR)) F2(F) --> XOR XOR --> E{{E}} </pre>	<pre> graph TD F1(F) --> AND((AND)) F2(F) --> AND AND --> E{{E}} </pre>	<pre> graph TD F1(F) --> OR((OR)) F2(F) --> OR OR --> E{{E}} </pre>



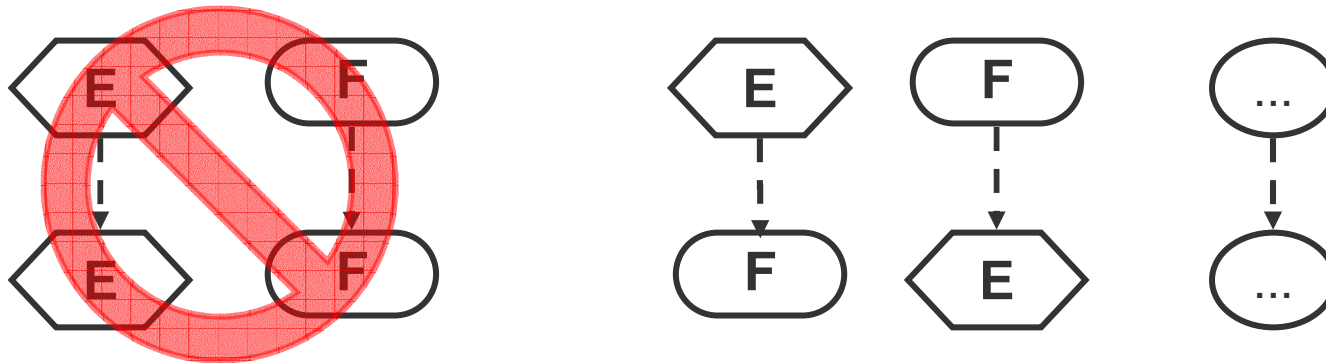


(1|6)

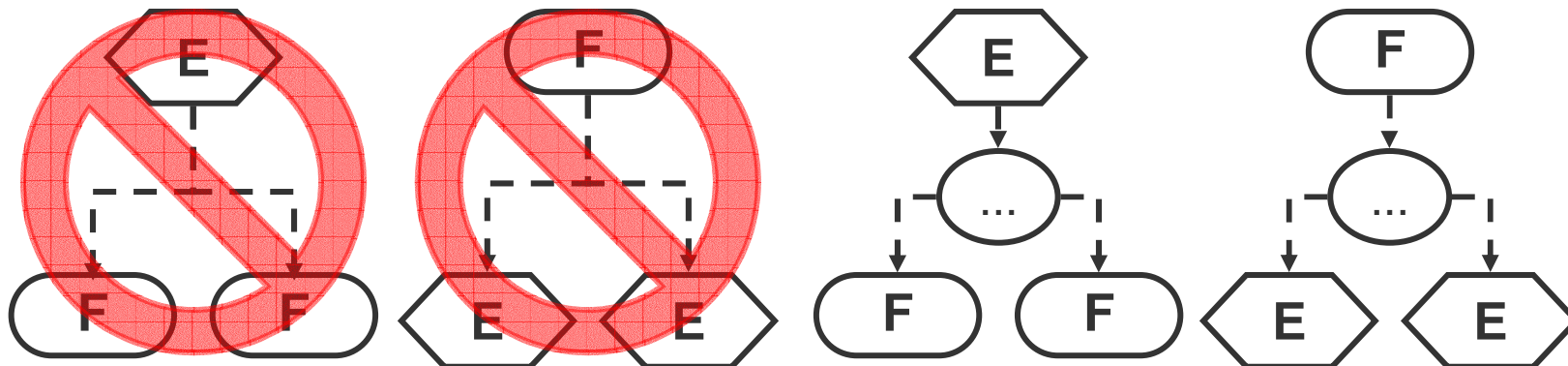
10.4 Grundregeln der EPK-Modellierung

Jede **Kante** verbindet **in der Regel** zwei Knoten von jeweils **unterschiedlichem** Typ.

Nur die Verknüpfungsoperatoren verzweigen, sie verbinden Ereignisse mit Funktionen und vice versa (sie können auch miteinander verbunden werden).



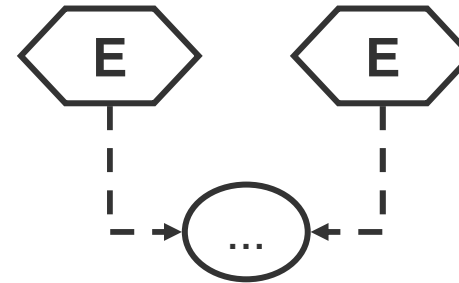
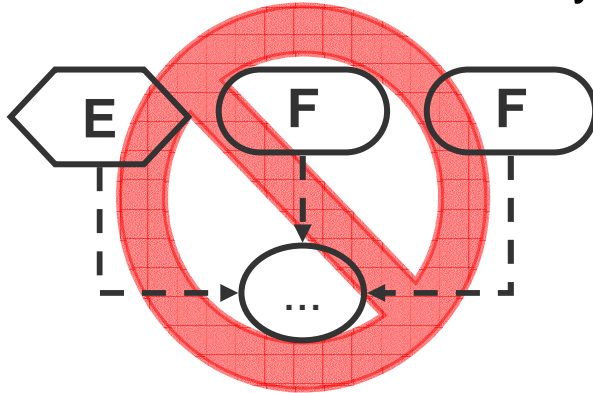
Verzweigung:



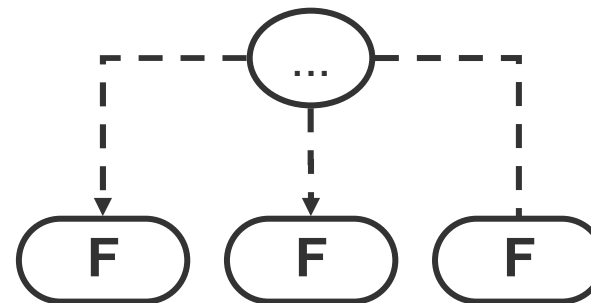
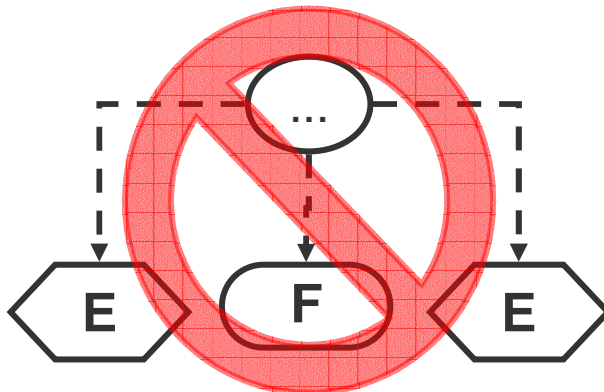
10.4 Grundregeln der EPK-Modellierung

(2|6)

Die **Eingänge eines Verknüpfungsoperators** sind entweder **alle vom** Typ Ereignis oder **alle vom** Typ Funktion,



ebenso sind **seine Ausgänge** entweder **alle vom** Typ Ereignis oder **alle vom** Typ Funktion.

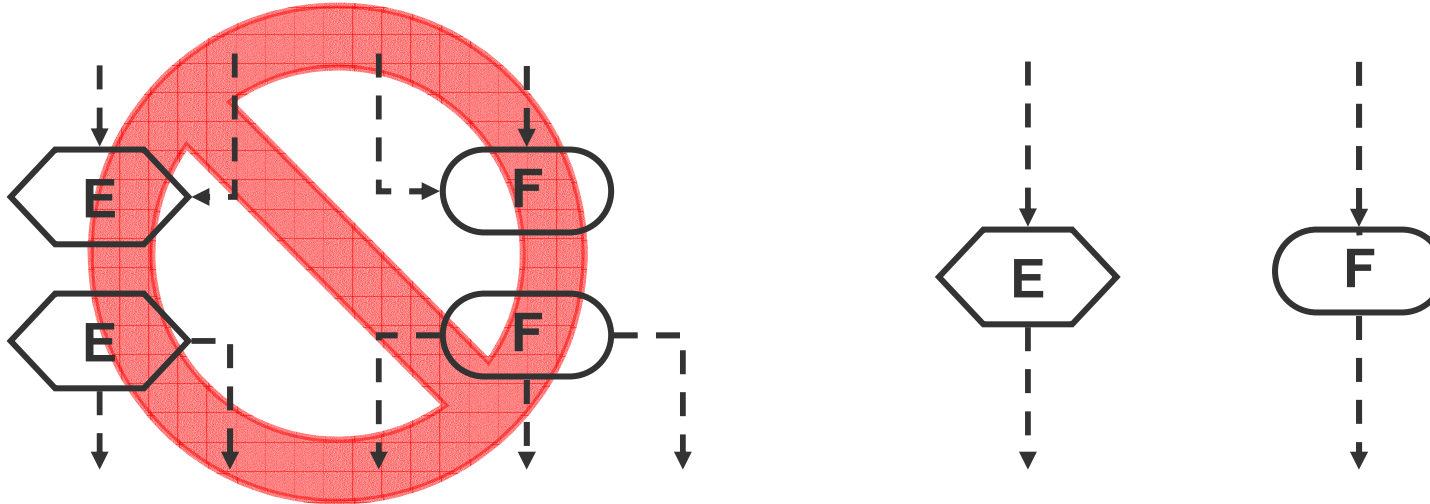


10.4 Grundregeln der EPK-Modellierung

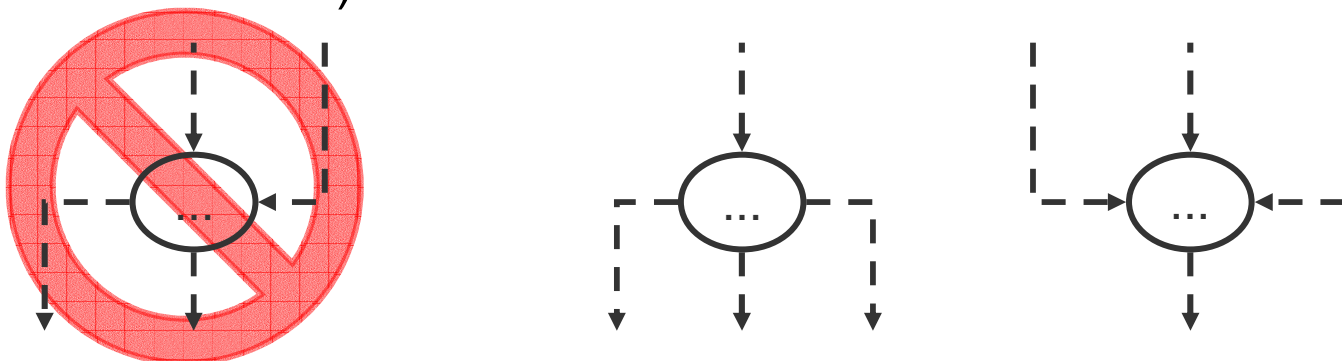
(3|6)

Ereignisse und Funktionen

dürfen nur **einen** Eingang und **einen** Ausgang haben;



Verknüpfungen können mehrere Eingänge oder Ausgänge haben (**aber nicht beides**).



10.4 Grundregeln der EPK-Modellierung

(4|6)

Verbot, auf ein Ereignis einen XOR- bzw. OR-Konnektor folgen zu lassen, da Ereignisse als passive Elemente keine Entscheidungen treffen können [Vgl. 10.3 – (8|8)]

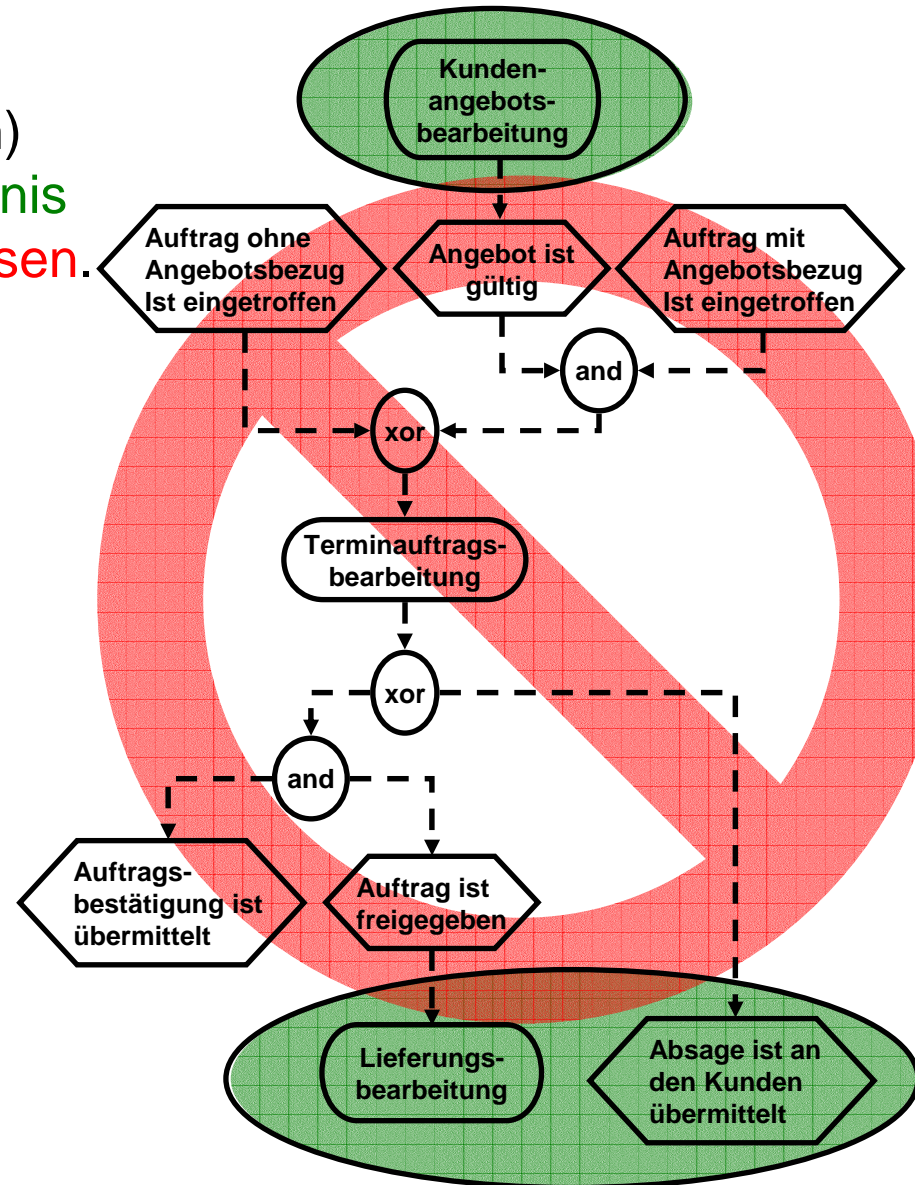
Operator Art	XOR entweder/oder	AND und	OR und/oder
Auslösendes Ereignis			
Erzeugtes Ereignis			



10.4 Grundregeln der EPK-Modellierung

(5|6)

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

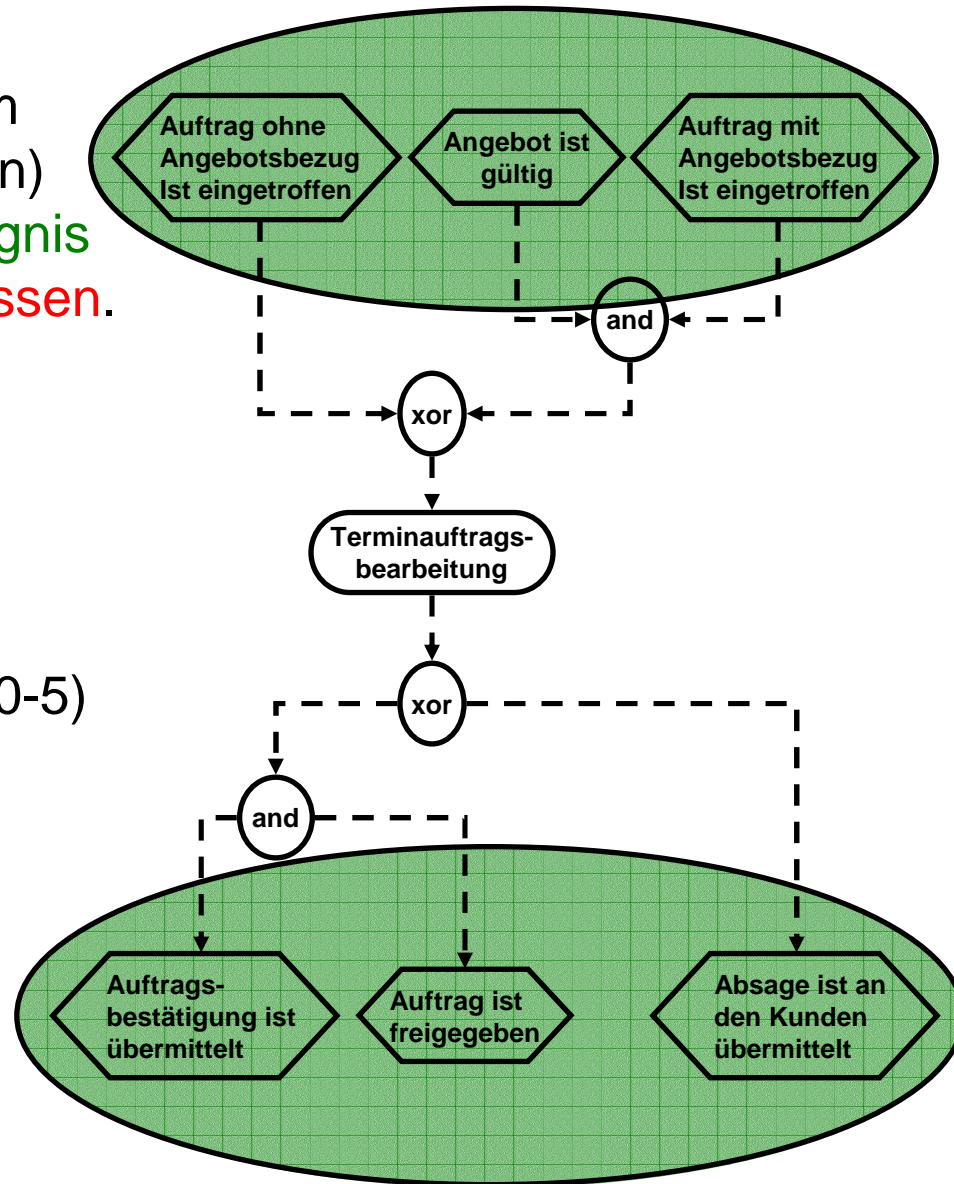


10.4 Grundregeln der EPK-Modellierung

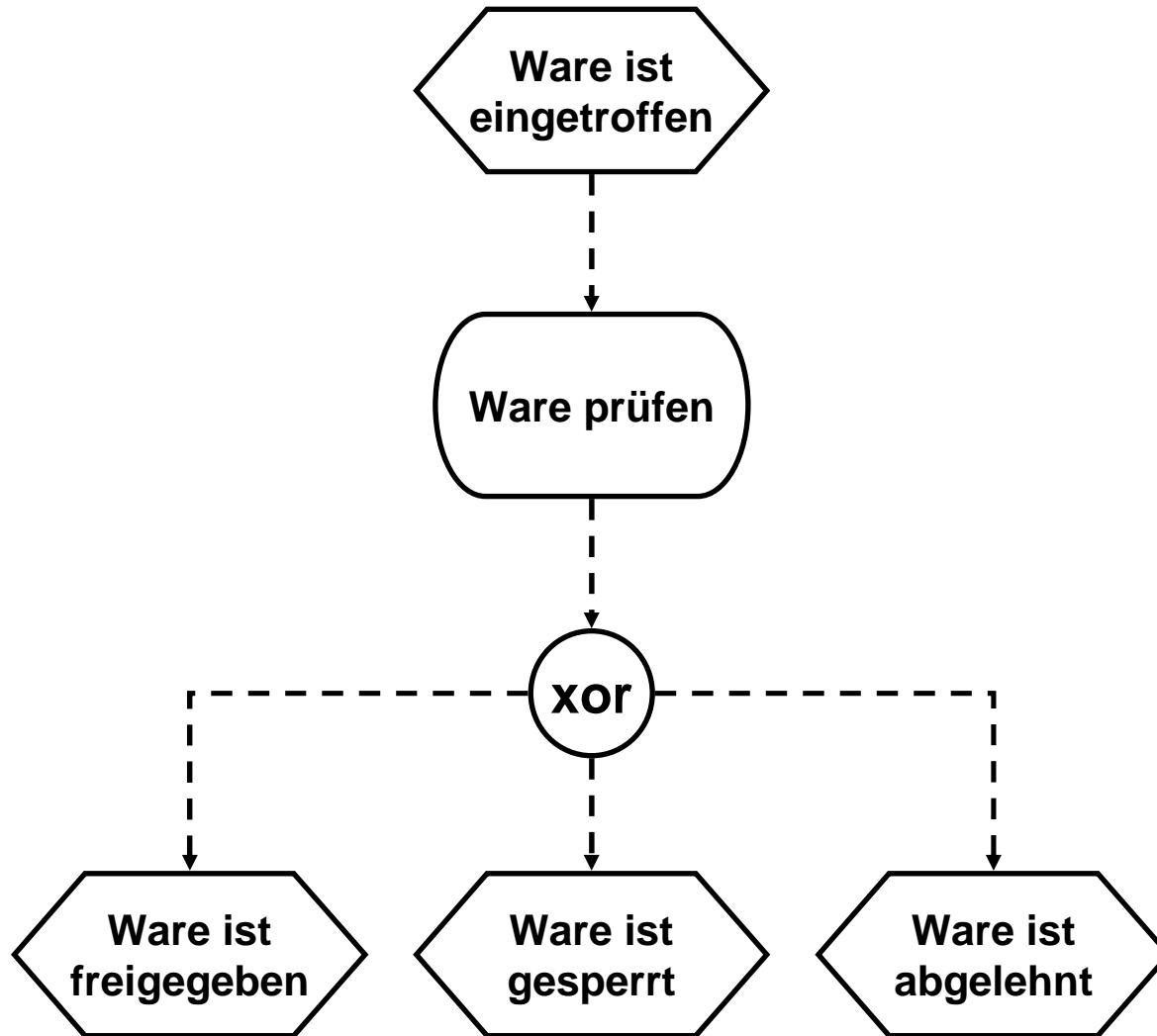
(6|6)

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

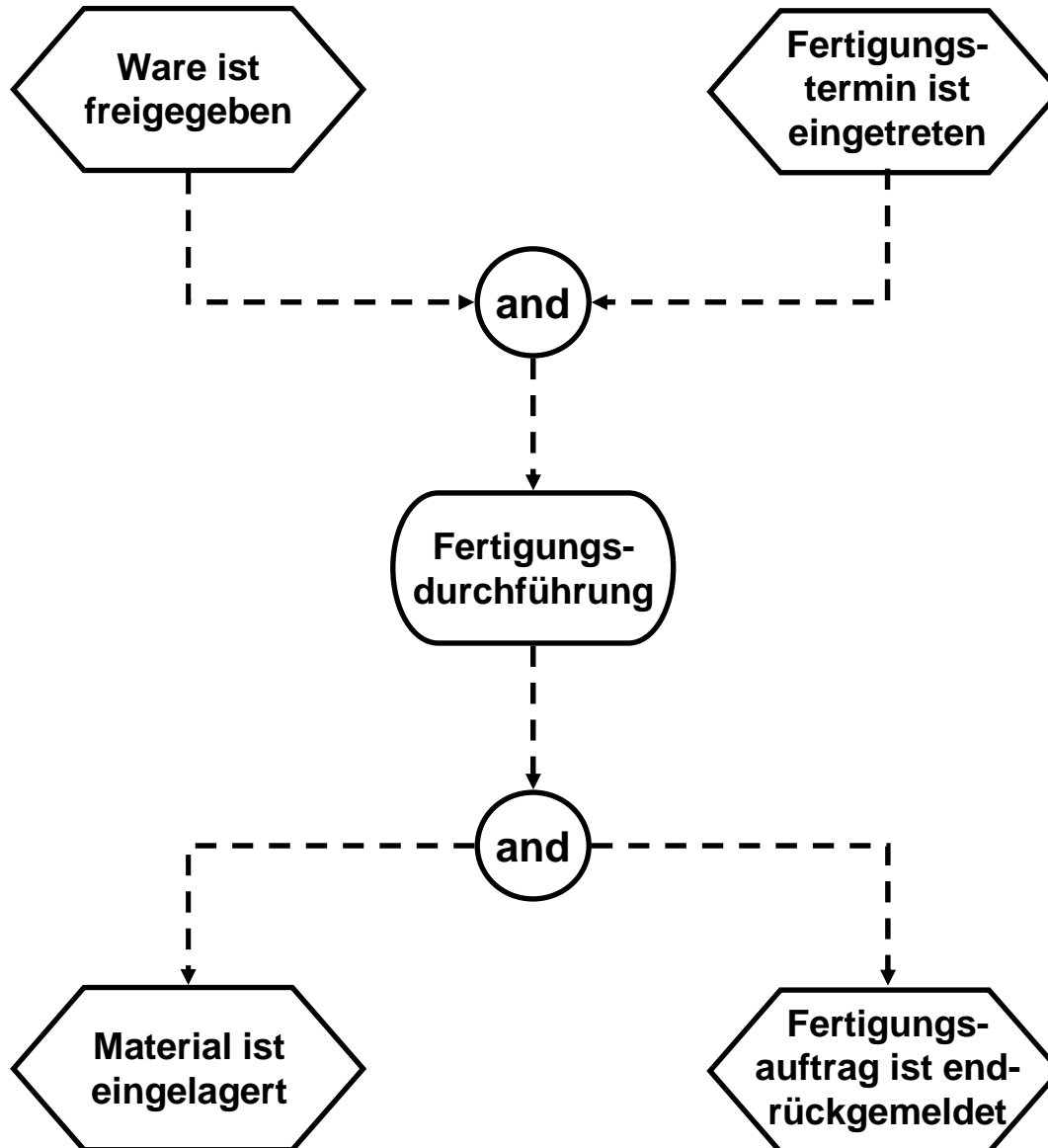
Ausnahme:
Verweis auf andere EPK
(siehe Beispiele 10-4 und 10-5)



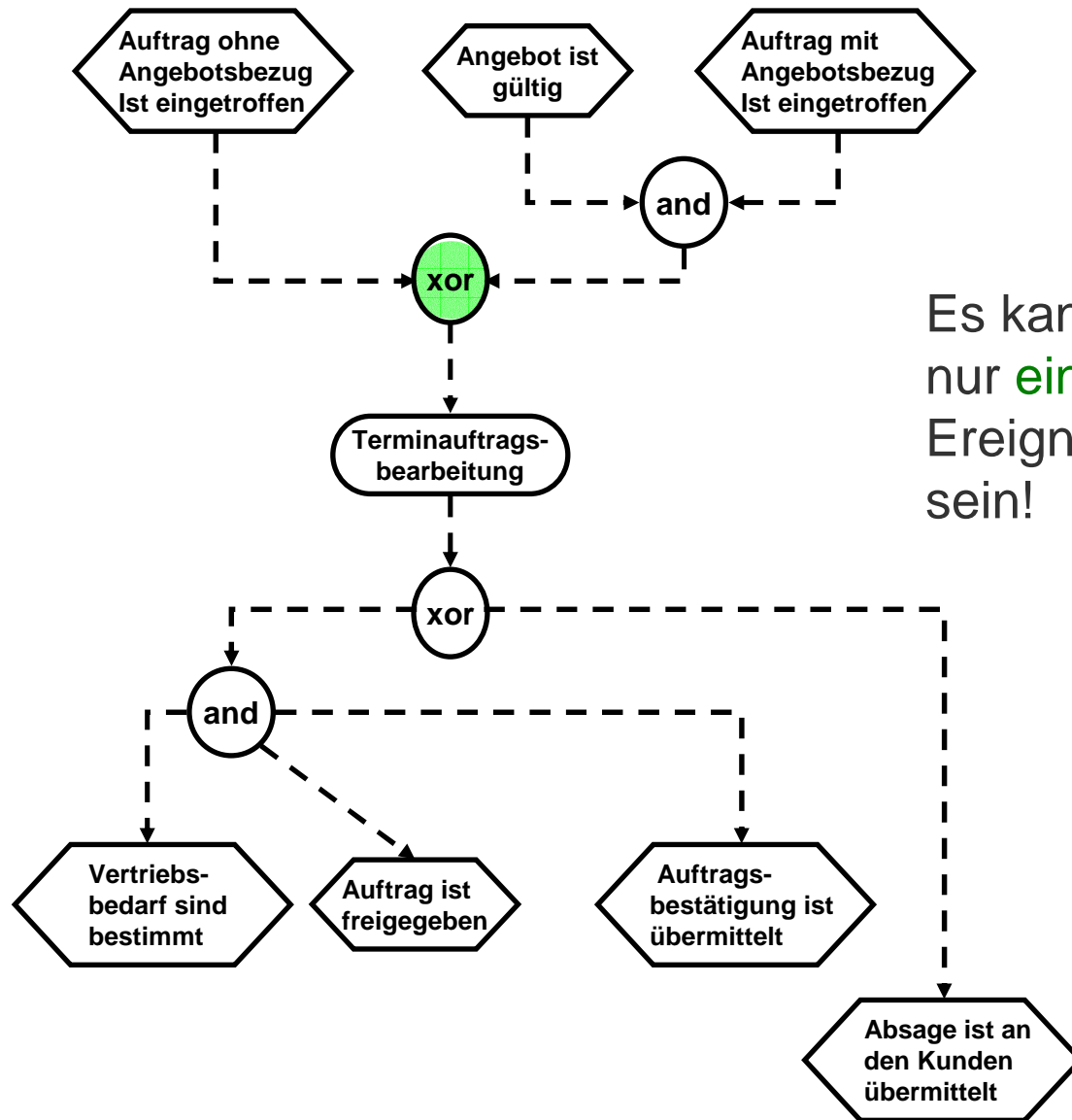
Beispiel 10-1: Wareneingangsbearbeitung



Beispiel 10-2: Fertigungsdurchführung



Beispiel 10-3: Auftragsbearbeitung

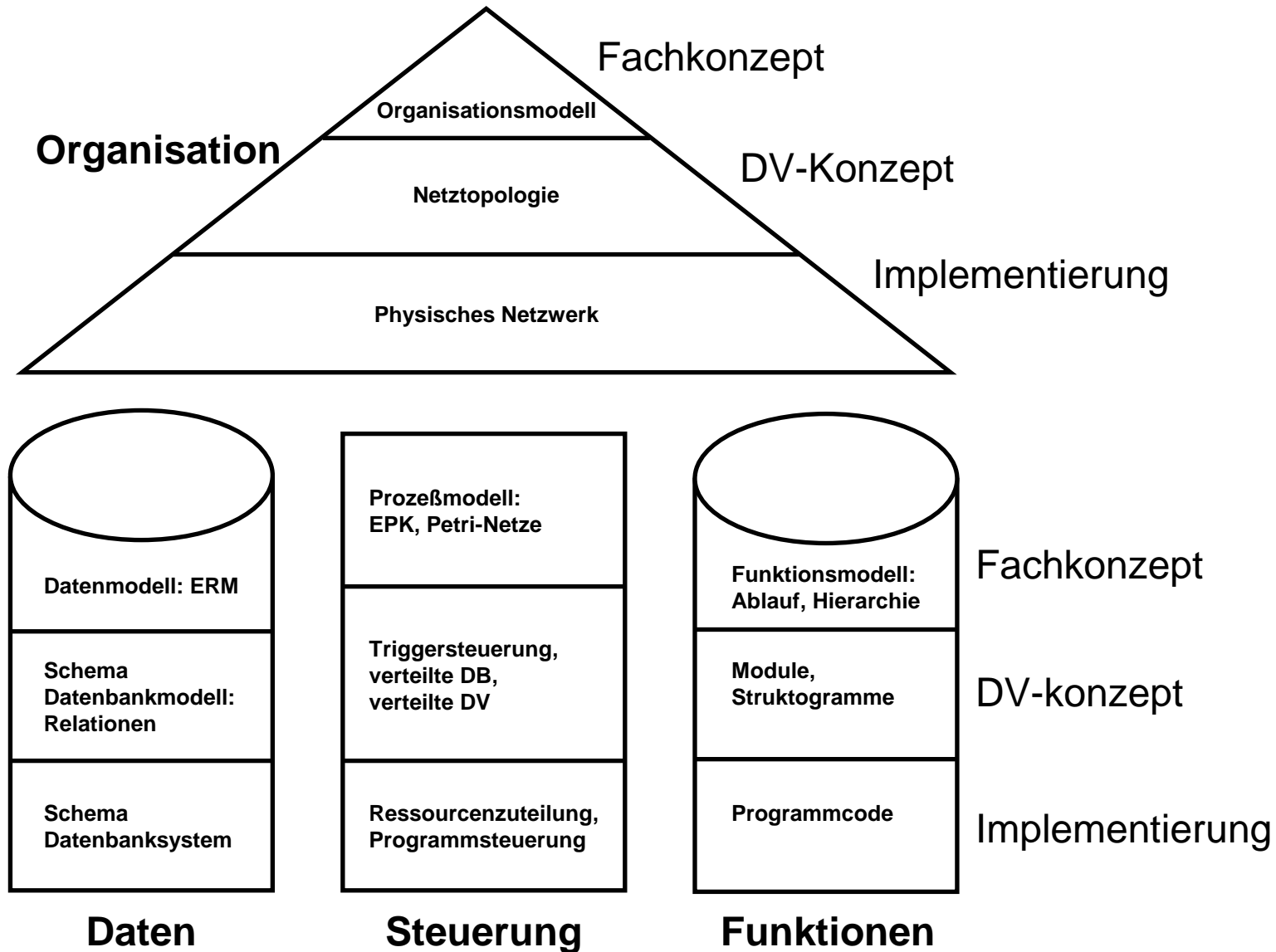


Es kann vernünftigerweise nur **eines** der beiden Ereignisse eingetreten sein!





10.5 ARIS (Architektur integrierter Informationssysteme)

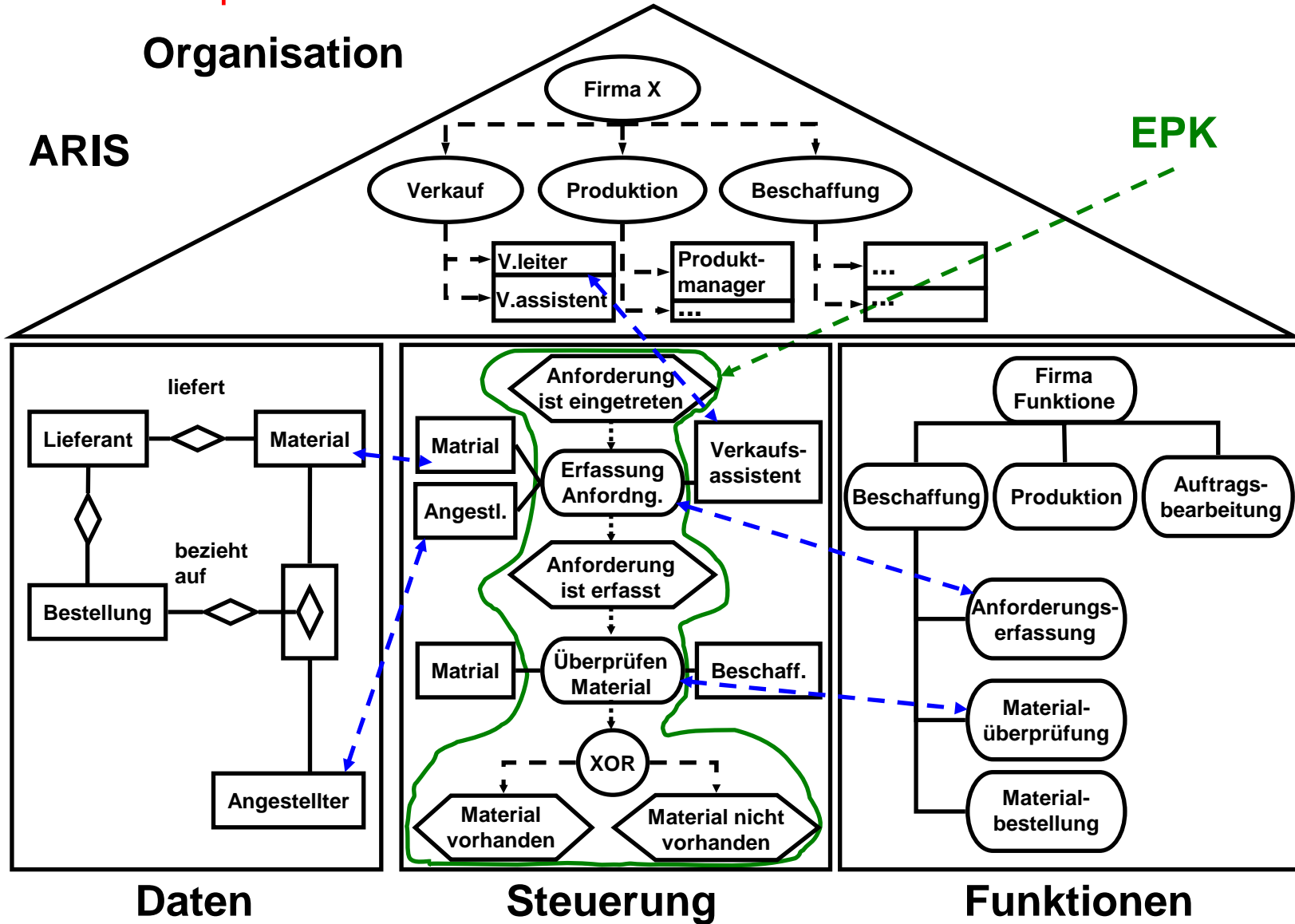




10.6 Erweiterte EPK und ARIS

(1|3)

Fachkonzept:



10.6 Erweiterte EPK und ARIS

(2|3)

Ergänzungselemente – Erweiterte EPK

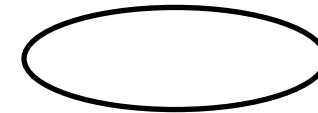
Daten- / Materialobjekt



Informations- / Materialfluß



Organisatorische Einheit



Zuordnung organisatorische Einheit



Funktion wird durch EPK verfeinert



10.6 Erweiterte EPK und ARIS

(3|3)

Den Funktionen können

- die mit der Ausführung betrauten Organisationseinheiten sowie
- ein- und ausgehende Datenobjekte **zugeordnet** werden.

Damit wird

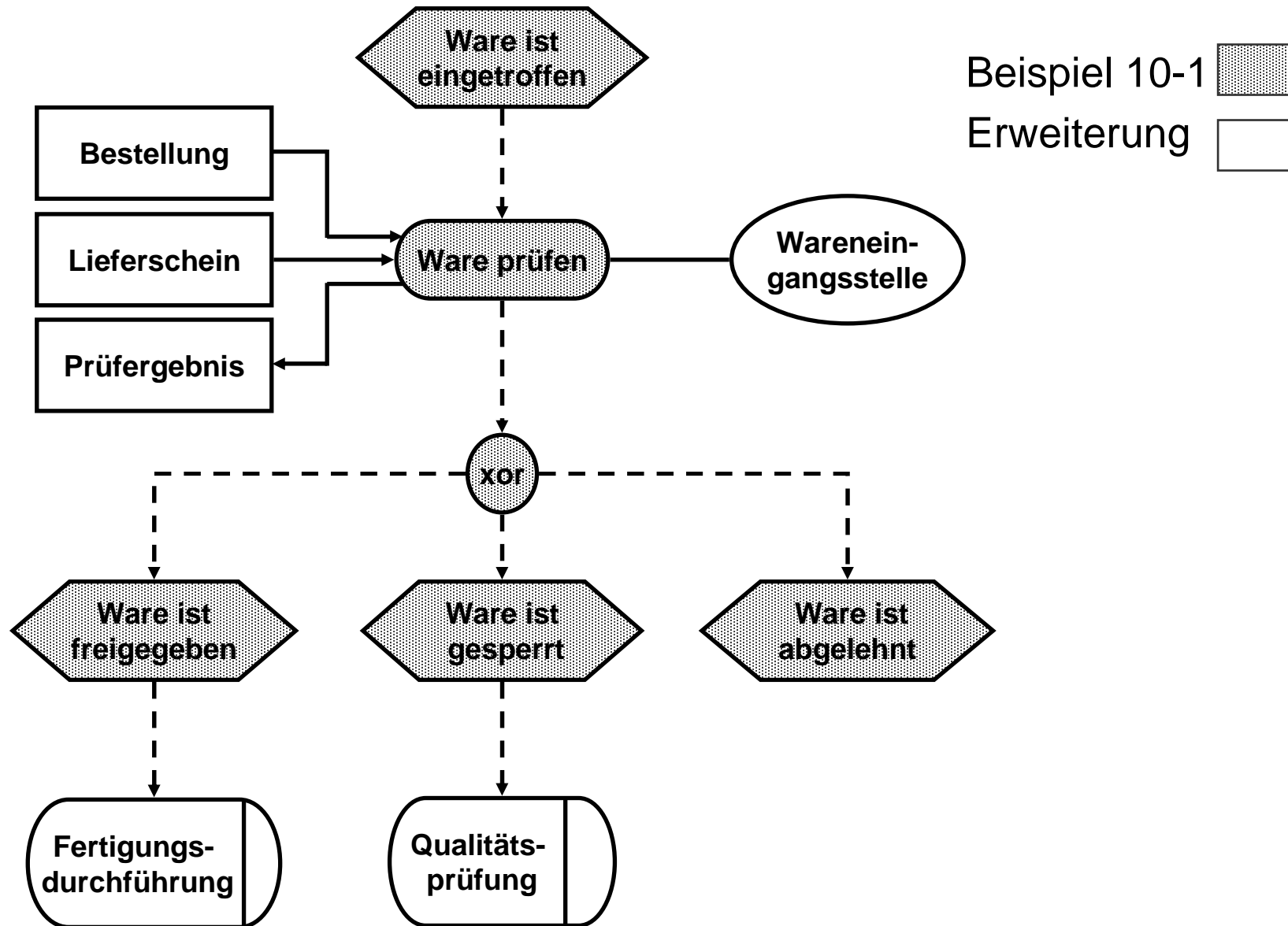
- die **Prozessverfolgung** über mehrere Organisationseinheiten hinweg möglich

und

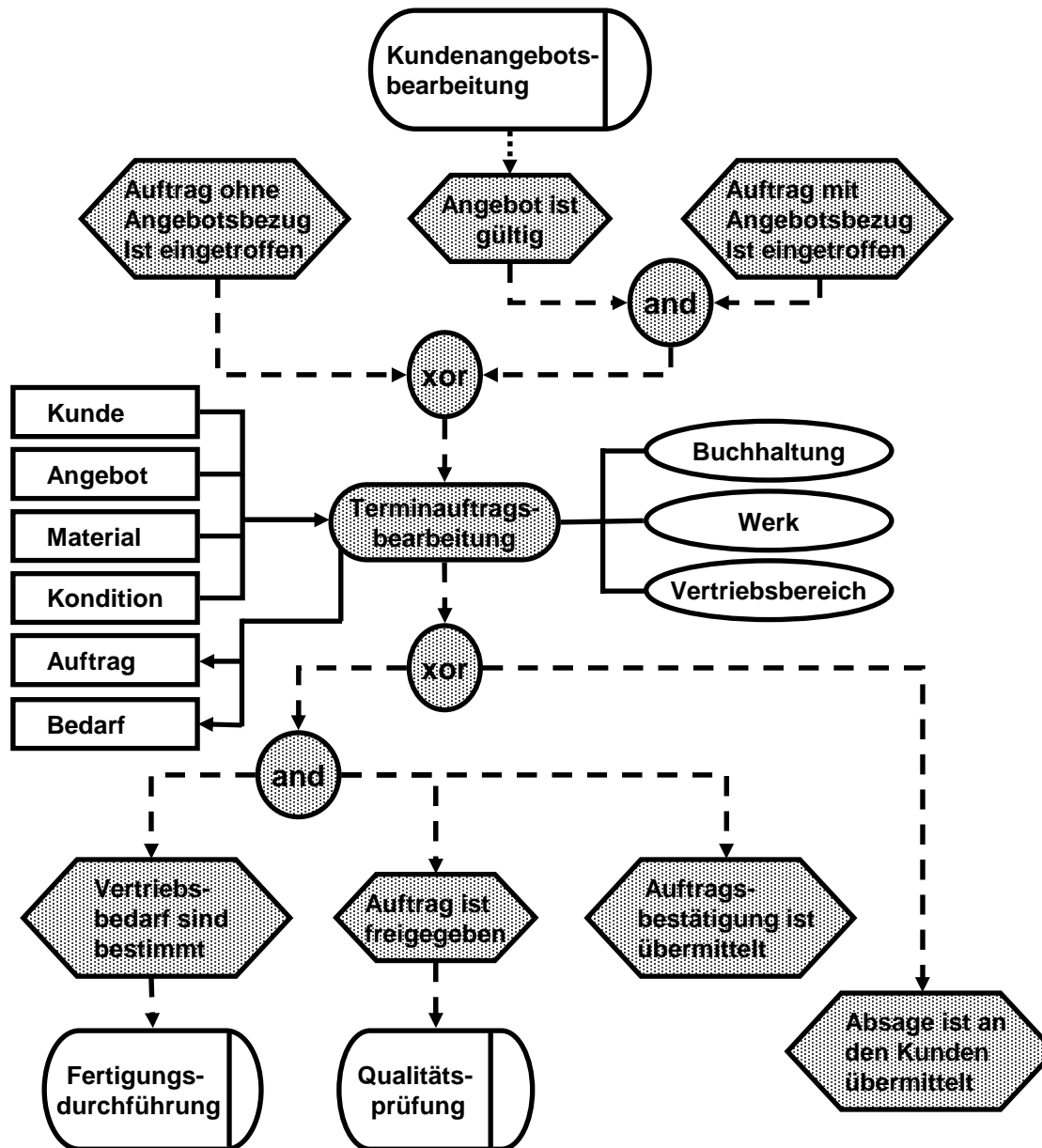
- **zu jedem Informationsobjekt** kann angegeben werden, von welcher organisatorischen Einheit es kommt, bzw. zu welcher organisatorischen Einheit es hingeschickt werden soll.



Beispiel 10-4: Wareneingangsbearbeitung



Beispiel 10-5: Auftragsbearbeitung



Beispiel 10-3 

Erweiterung 





(1|8)

10.7 EPK - Petrinetze

(nach van der Aalst: 1998)

Die EPK-Methode

- basiert auf Petrinetztheorie,
- kann verstanden werden als eine Variante des Stellen/Transitions-Netzes, welches um logische Verknüpfungsoperatoren erweitert wurde,
- bietet wie die Petrinetze eine prozessorientierte Betrachtung der Daten- und Funktionssicht (*ARIS*-Steuerungssicht).



10.7 EPK - Petrinetze

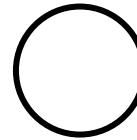
(2|8)

EPK

Petrinetze



Ereignis



Stelle



Funktion



Transition



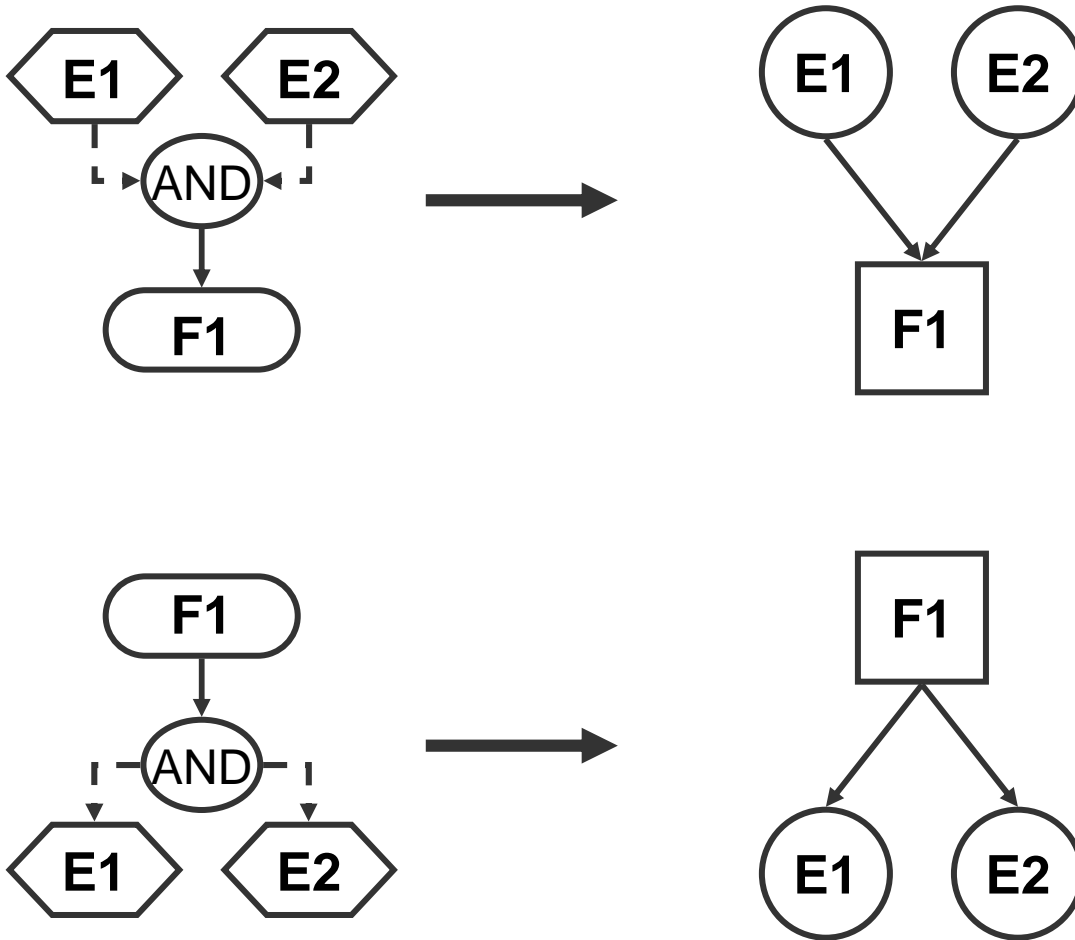
?



10.7 EPK - Petrinetze

(3|8)

Übersetzung von AND in Petrinetze



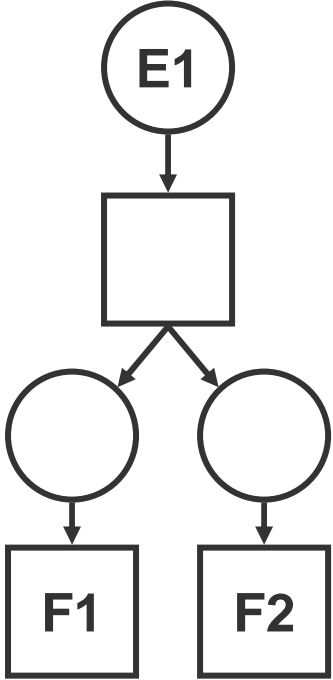
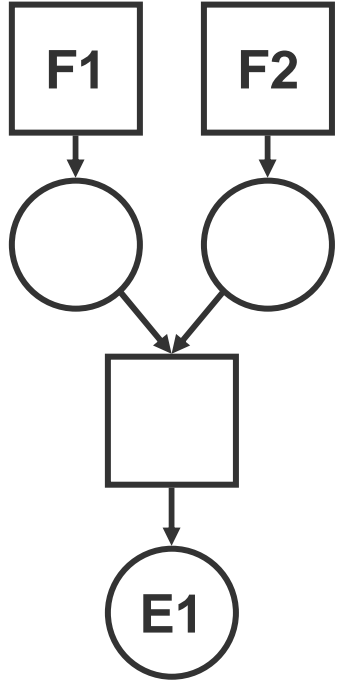
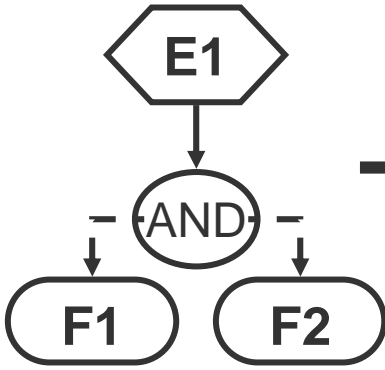
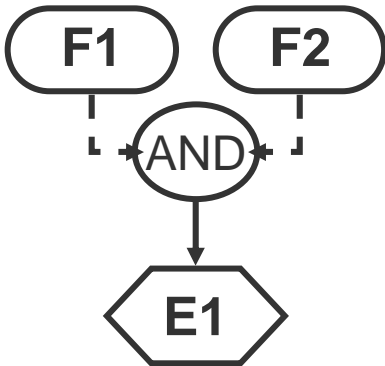
27.01.2003



10.7 EPK - Petrinetze

(4|8)

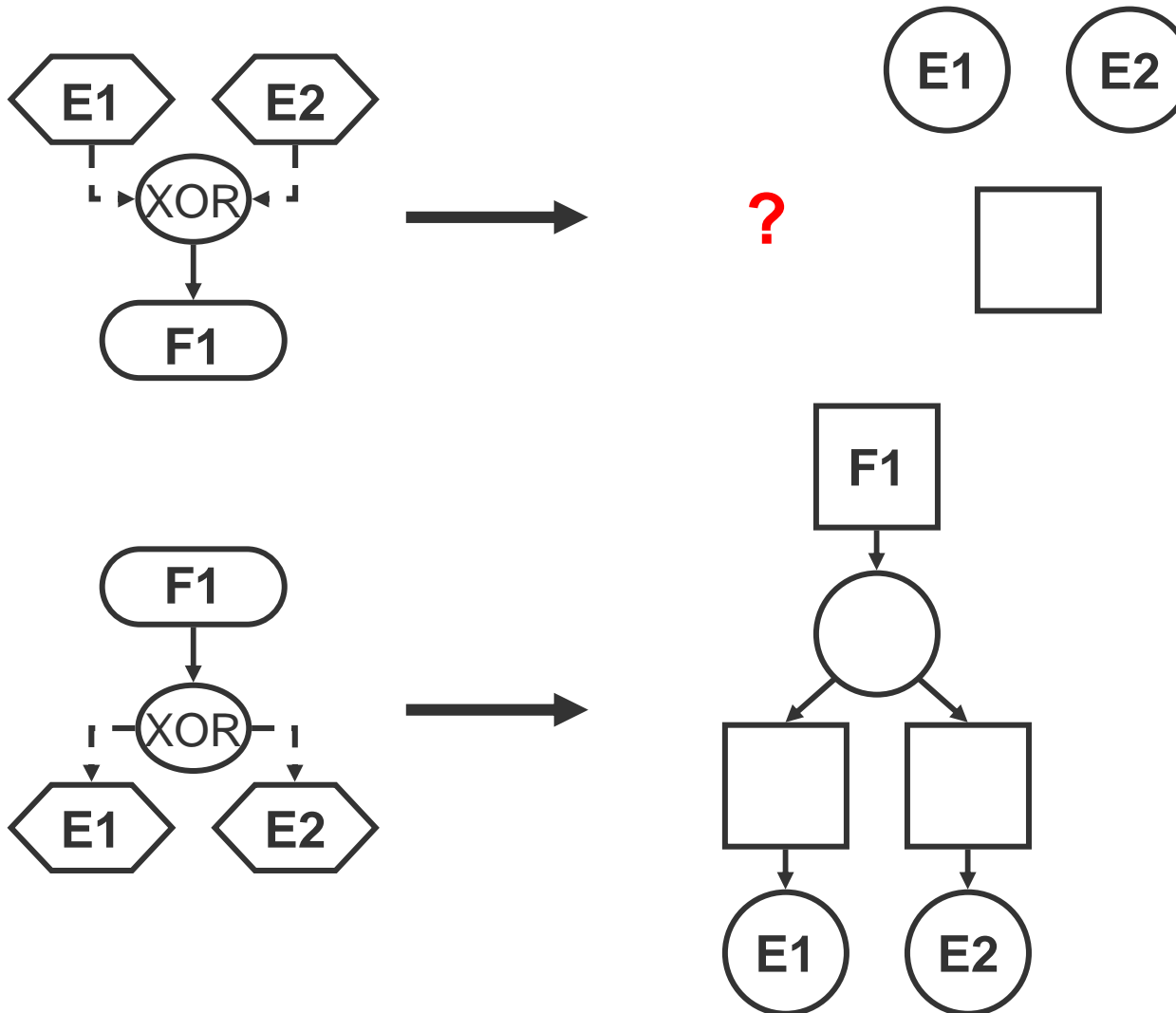
Übersetzung von AND in Petrinetze



10.7 EPK - Petrinetze

(7|8)

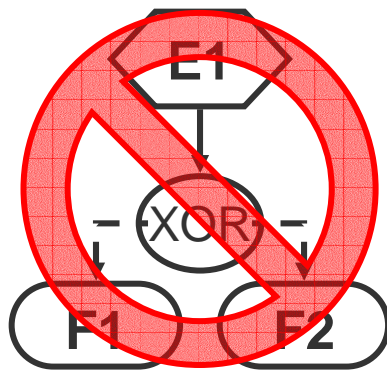
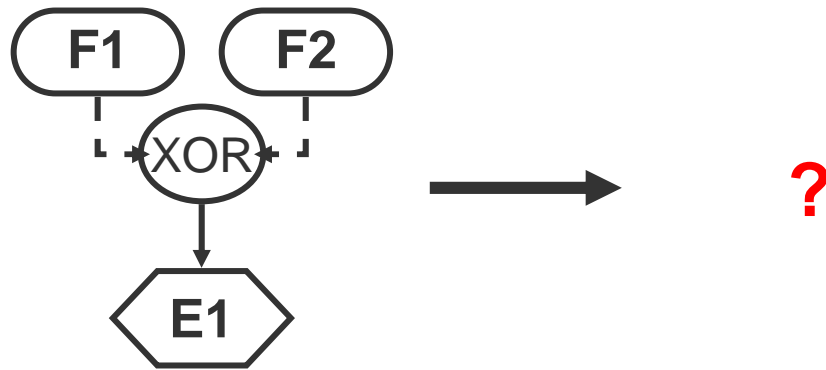
Übersetzung von XOR in Petrinetze



10.7 EPK - Petrinetze

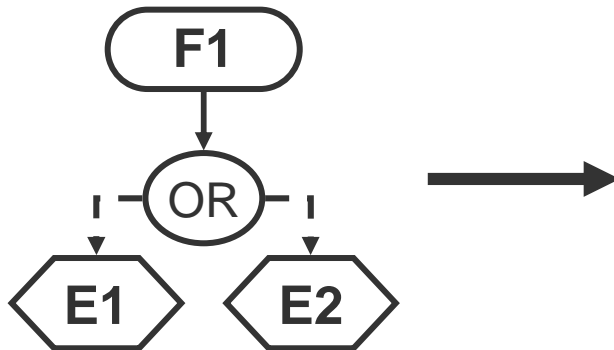
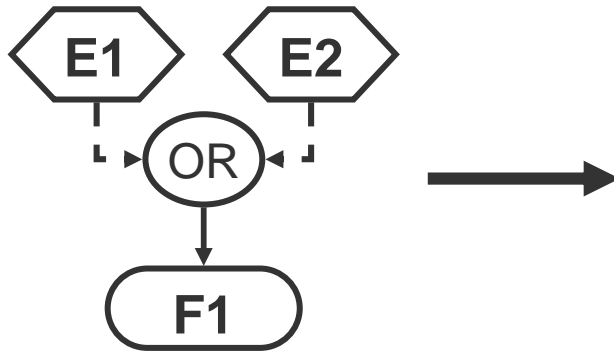
(8|8)

Übersetzung von XOR in Petrinetze



10.7 EPK - Petrinetze

Übersetzung von OR in Petrinetze



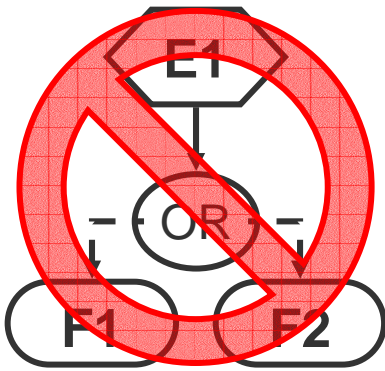
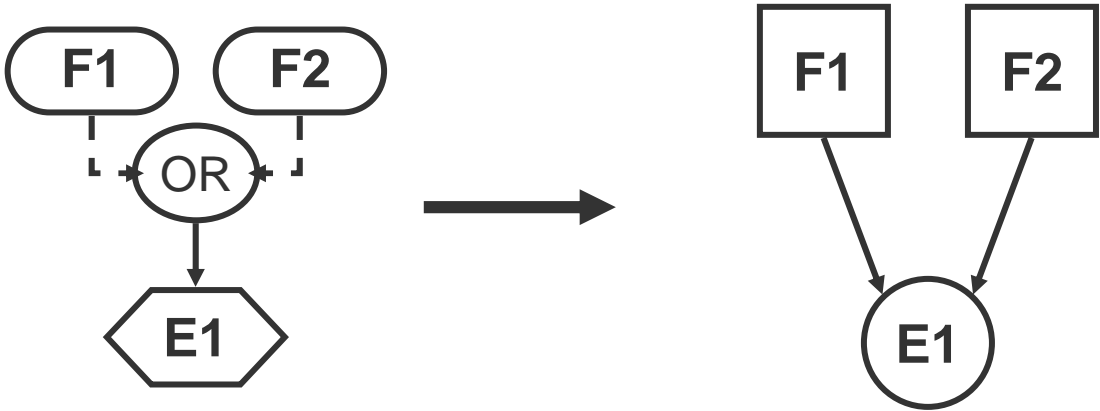
(5|8)



10.7 EPK - Petrinetze

(6|8)

Übersetzung von OR in Petrinetze





10.8 Bewertung EPK im Hinblick auf Petrinetze

- hohe Anschaulichkeit und Sichtintegrationsfähigkeit (ARIS)
- nur eine statische Sicht auf Prozessstrukturen
- Während sich Petrinetze sehr gut als Simulationssprache verwenden lassen, können die EPK diesen erhöhten methodischen Anforderungen ohne syntaktisch-semantic 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.



Literatur

W.M.P. van der Aalst: Formalization and Verification of Event-driven Process Chains. Information and Software Technology 41, S. 639–650, 1999.

<http://citeseer.nj.nec.com/vanderaalst99formalization.html>

W. van der Aalst, J. Desel, E. Kindler: On the semantics of EPCs: A vicious circle.
In: M. Nüttgens, F.J. Rump: EPK 2002 - Geschäftsprozessmanagement mit Ereignisgesteuerten Prozessketten, S. 71–79.

M. Nüttgens, F.J. Rump: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In: Promise 2002 - Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen, Proceedings des GI-Workshops und Fachgruppentreffens (Potsdam, Oktober 2002), LNI Vol. P-21, Bonn 2002, S. 64–77.

P. Langner, C. Schneider, J. Wehler: Prozeßmodellierung mit ereignisgesteuerten Prozeßketten (EPKs) und Petri-Netzen. Wirtschaftsinformatik 39, S. 479–489, 1997.

P. Rittgen: Quo vadis EPK in ARIS? Ansätze zu syntaktischen Erweiterungen und einer formalen Semantik. Wirtschaftsinformatik 42, S. 27–35, 2000.

