

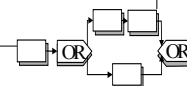
## Petri-Netze: Analyse

### ■ Bzgl. der formalen Analyse von Petri-Netzen

- Erreichbarkeitsanalyse
- Deadlock-Analyse
- Lebendigkeit
- Invarianten

### ■ wird verwiesen auf

- Baumgarten, B.: *Petri-Netze: Grundlagen und Anwendungen*. Spektrum Akademischer Verlag, Heidelberg, 1996.
- Reisig, W.: *Petrinetze: eine Einführung*. Springer, Berlin, 1991.
- Starke, P.: *Analyse von Petri-Netz-Modellen*. Teubner, Stuttgart, 1990.
- Petri-Netz-Vorlesungen



## Petri-Netze: Bewertung von Prädikat-Transitions-Netzen bzgl. Workflow-Definitionen

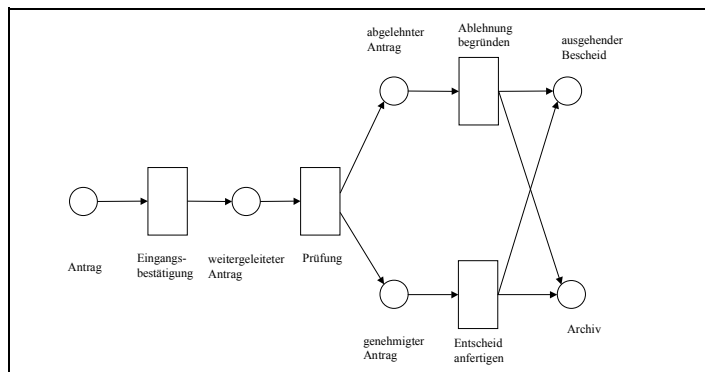
### ■ Prädikat-Transitions-Netze (Pr/T) nutzbar für Workflow-Definitionen, da

- Unterscheidbare Tokens, Kantengewichte
- Prädikate an Stellen und Transitionen für zusätzliche Steuerung des Schaltverhaltens

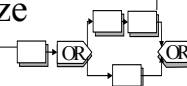
### ■ Problematik: Kontrollfluss muss implizit durch Datenfluss spezifiziert werden

### ■ Limitationen von Pr/T-Netzen u.a.

- Kein Typkonzept für Tokens (im programmiersprachlichen Sinn)
- Keine Zugriffsstrategien spezifizierbar bzgl. Stellen (FIFO, LIFO, ...)
- Keine Differenzierung des Schaltverhaltens von Transitionen
- Keine Spezifikation möglich, ob Arbeitsschritt manuell oder automatisch
- Keine temporale Unterstützung

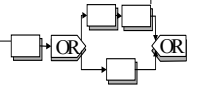


### ■ Stellvertreter für workflow-orientierte Petri-Netz-Typen: FUNSOFT-Netze

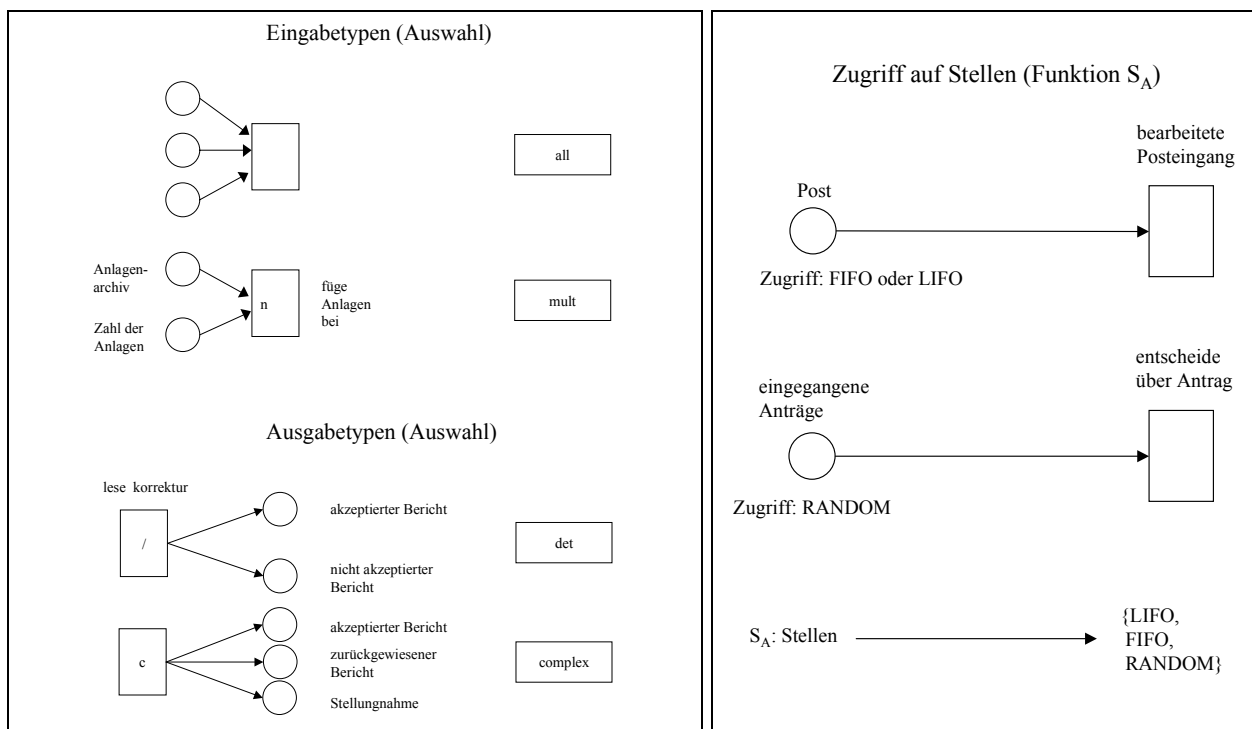


## FUNSOFT-Netze: Hauptmerkmale

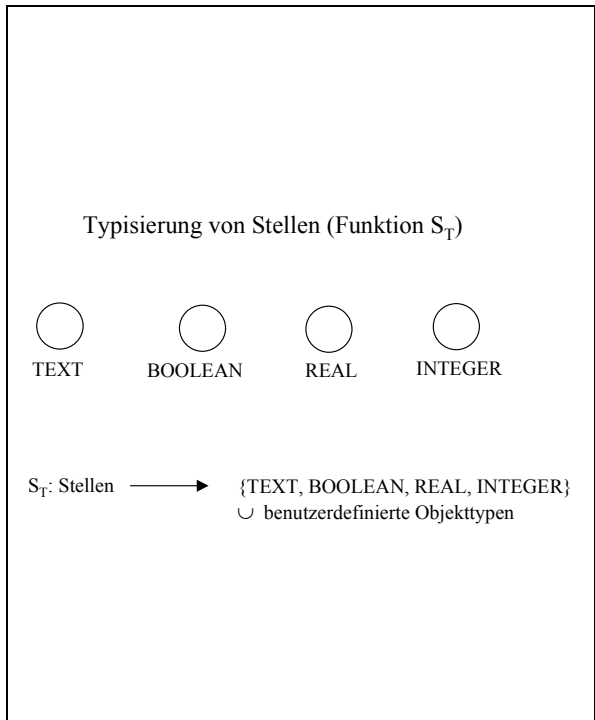
- Beispiel eines workflow-orientierten Petri-Netz-Typs
- Entwickelt von der Fraunhofer-Gesellschaft (Deiters et al.)
- Token-Typisierung möglich
- Unterschiedliche Zugriffsstrategien auf Stellen (Kanäle)
- Unterschiedliches Schaltverhalten von Transitionen spezifizierbar
- Aktivitäten (modelliert durch Transitionen) können Attribute haben wie z.B.
  - Zeitverbrauch
  - Ausführungsmodus (manuell, automatisch)
  - Anzahl simultaner Ausführungen
  - Verfeinerungsmodus
- Formale Semantik durch definierte Abbildung auf Pr/T-Netze
- Werkzeuge CORMAN und LEU



## FUNSOFT-Netze: Schaltverhalten und Stellenzugriff

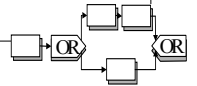


# FUNSOFT-Netze: Typisierung von Stellen und Tokens (Objekten)

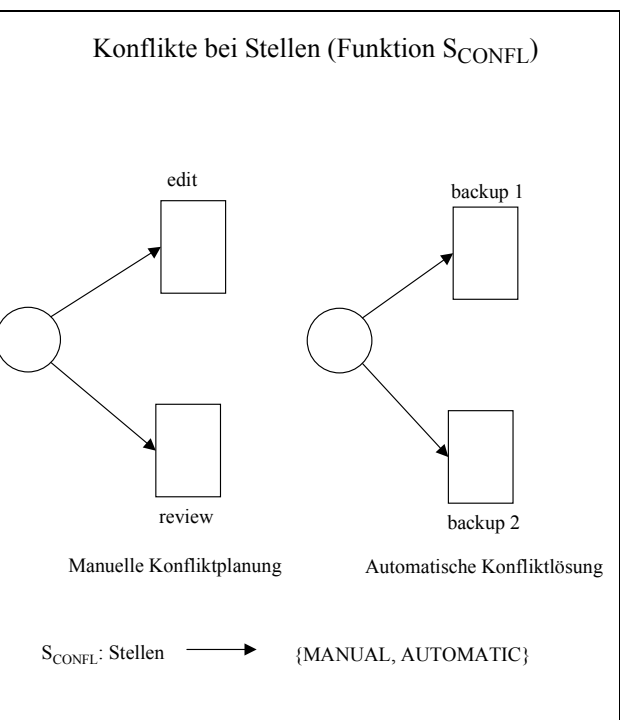
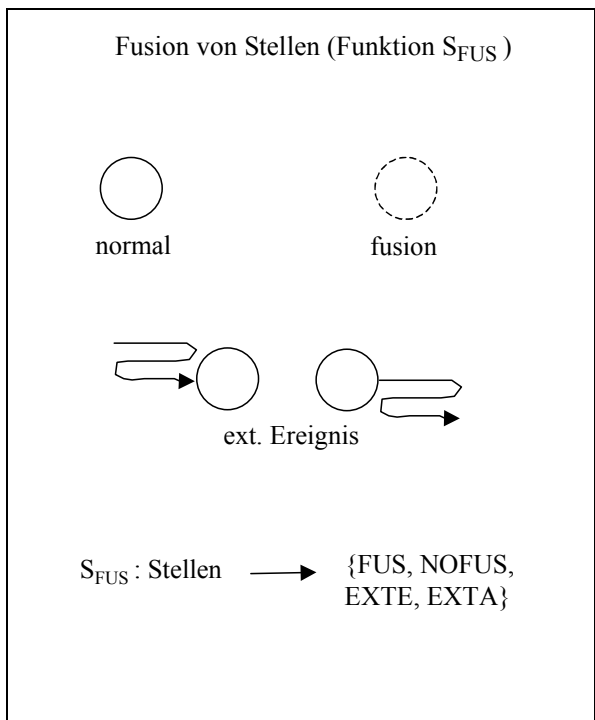


Benutzerdefinierte Typen wie z.B.:

```
OBJECT-TYPE-NAME = BERICHT
struct{
    string name;
    string titel;
    string autor;
    date datum-der-erstellung;
    TEXT inhalt;
    text glossar;
}
```

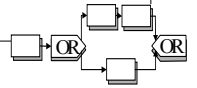


# FUNSOFT-Netze: Stellenfusion und Konfliktlösung



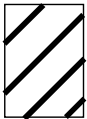
## FUNSOFT-Netze: Stellen (Zusammenfassung)

Funktion	Bedeutung	Domain	Wertebereich
$S_A$		Stellen	{LIFO, FIFO, RANDOM}
$S_T$		Stellen	Typen
$S_{FUS}$		Stellen	{FUS, NOFUS, EXTE, EXTA}
$S_{CONFL}$		Stellen	{MANUAL, AUTOMATIC}



## FUNSOFT-Netze: Verfeinerung von Transitionen

bearbeitete Antrag



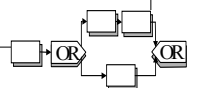
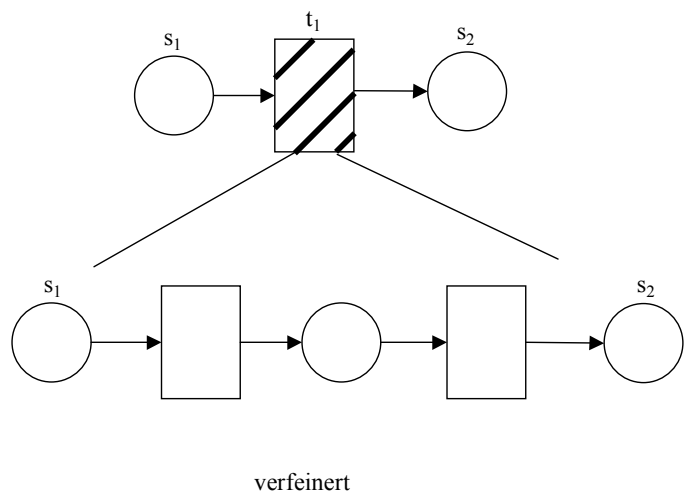
verfeinert

verbessere - Bericht

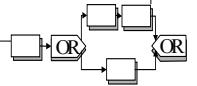
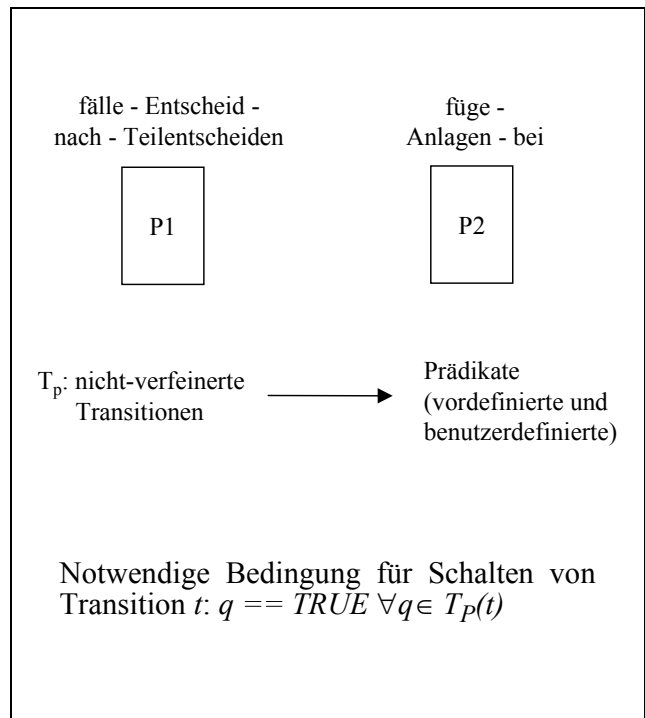
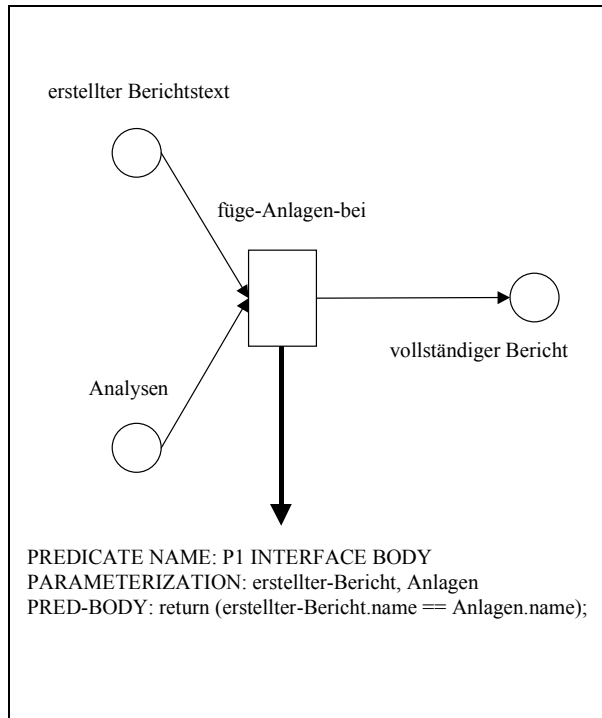


nicht verfeinert

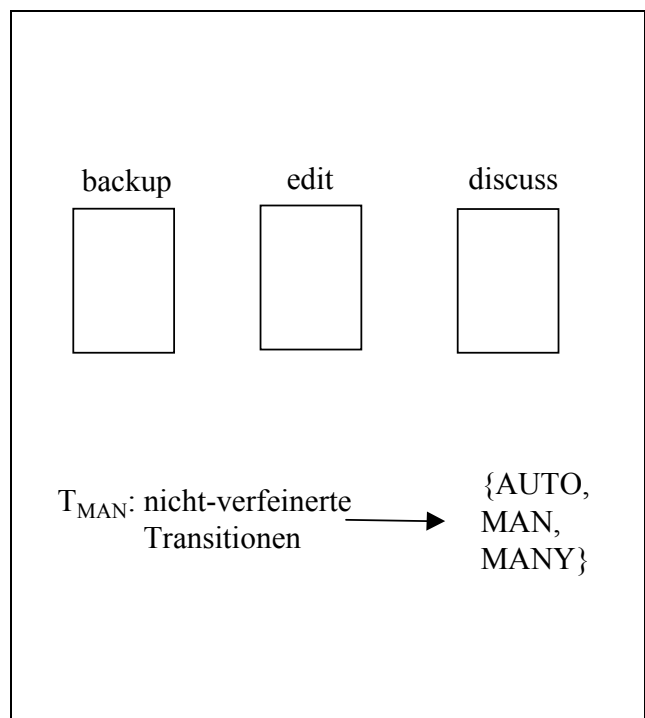
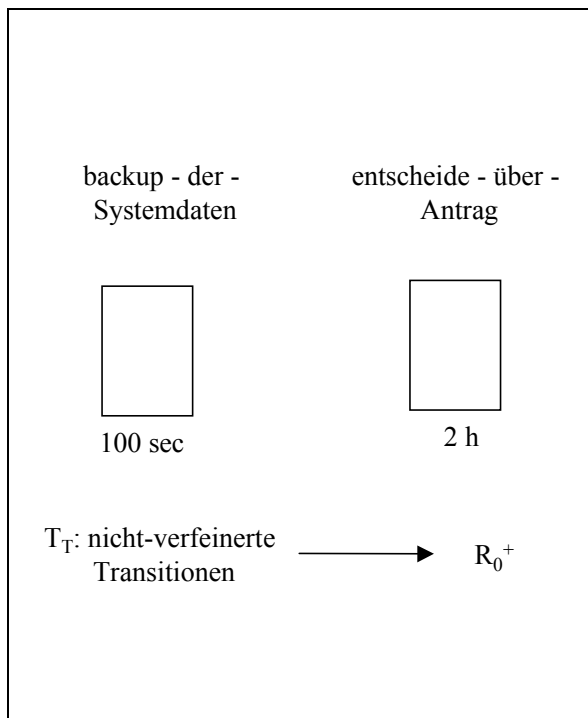
$T_{DEC}$ : Transitionen  $\longrightarrow$  {DEC, NODEC}



## FUNSOFT-Netze: Transitions-Prädikate

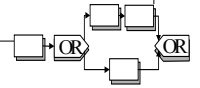


## FUNSOFT-Netze: Zeitverbrauch und Automation

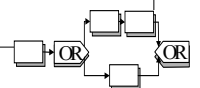
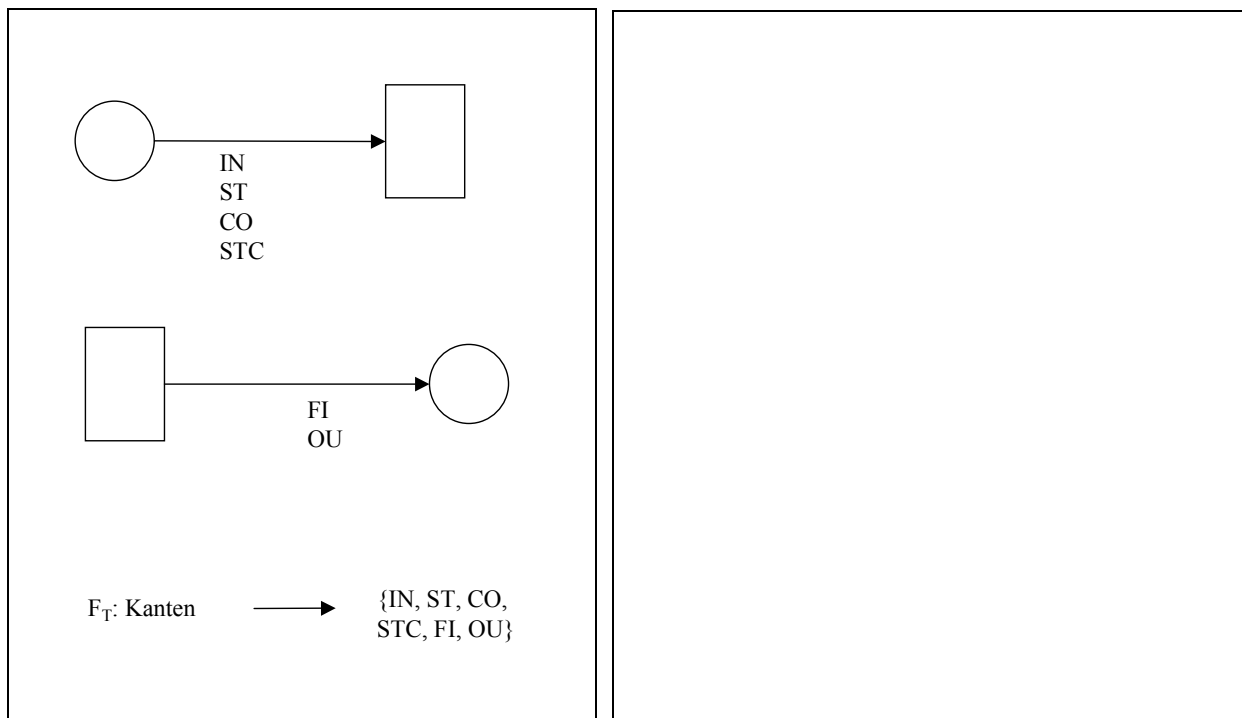


## FUNSOFT-Netze: Transitionen (Zusammenfassung)

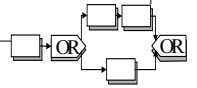
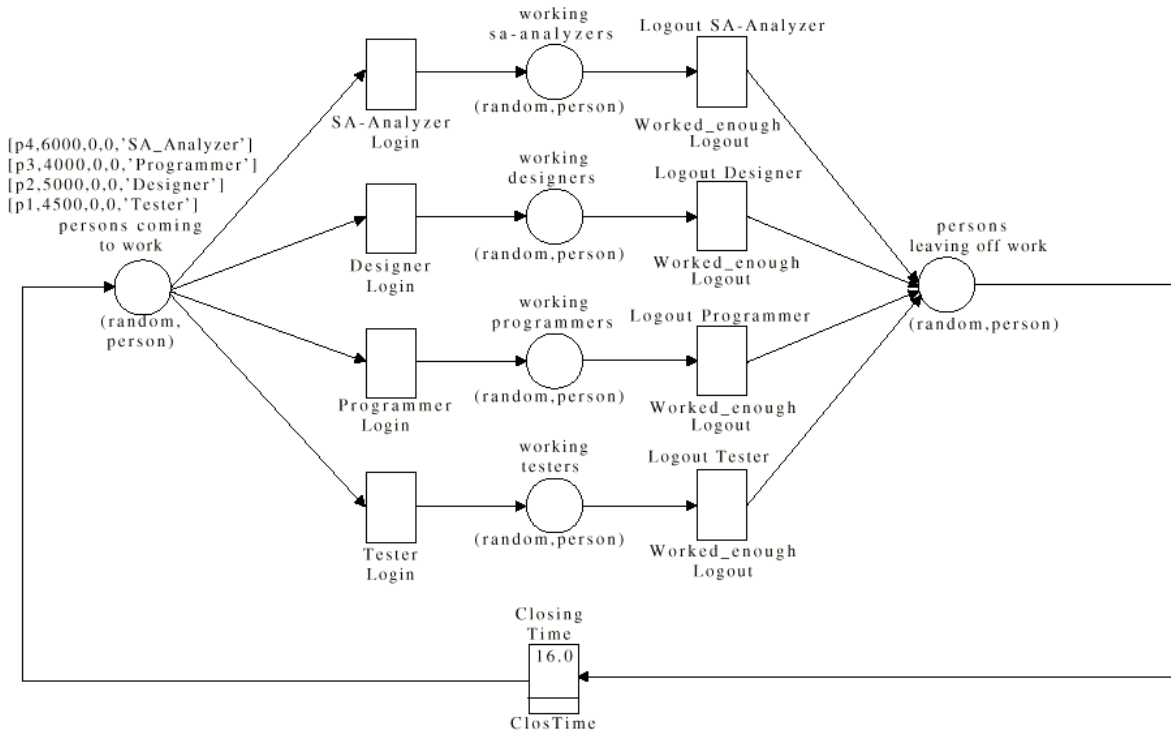
Funktion	Bedeutung	Domain	Wertebereich
$T_{DEC}$		Transitionen	{DEC, NODEC}
$T_J$		nicht-verfeinerte Transitionen	Jobs
$T_P$		nicht-verfeinerte Transitionen	Prädikate
$T_T$		nicht-verfeinerte Transitionen	$R^+_0$
$T_{MAN}$		nicht-verfeinerte Transitionen	{AUTO, MAN, MANY}



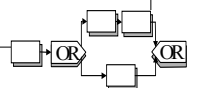
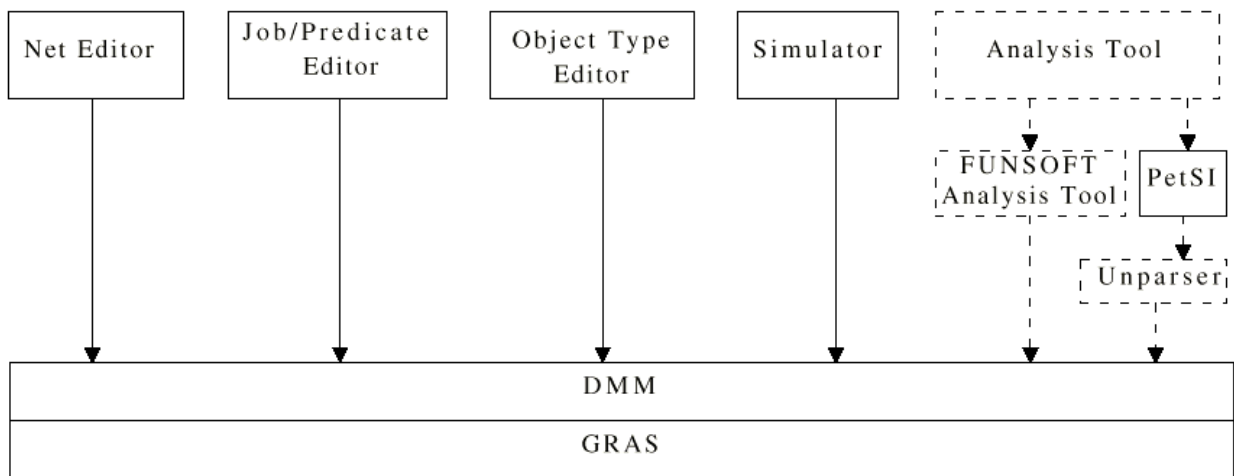
## FUNSOFT-Netze: Kantentypen



## FUNSOFT-Netze: Beispiel

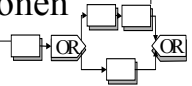


## FUNSOFT-Netze: Werkzeuge

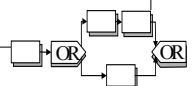
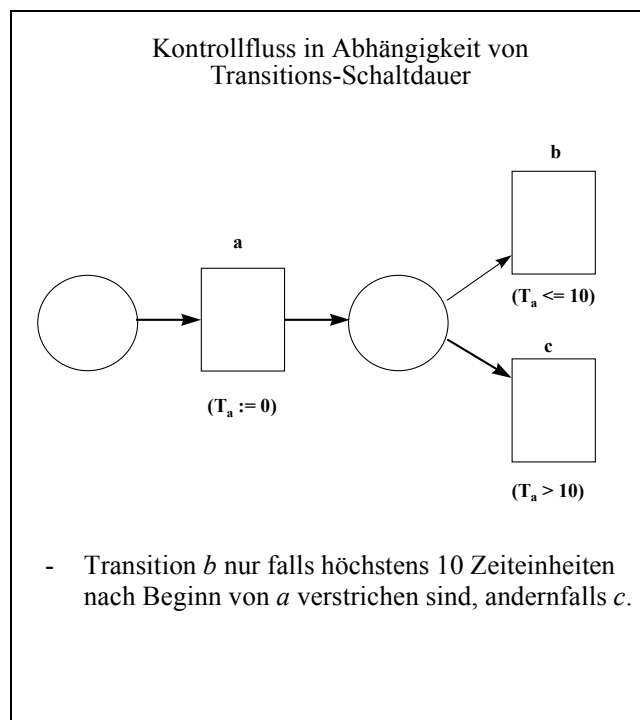
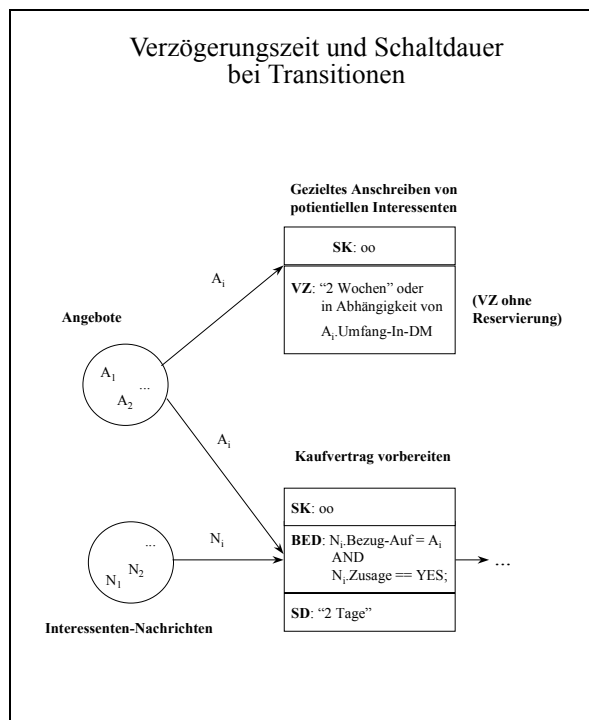


## Zeitorientierte Petri-Netze (1)

- Explizite Modellierung von temporalen Aspekten
- Mindestverweildauer von Tokens in Stellen
  - Reservierung eines Tokens durch aktivierte Transition  $t$  (Schaltung von  $t$  garantiert, aber Verzögerung von  $t$  um Mindestverweildauer bzgl. Stelle)
  - Keine Reservierung von Tokens; nach Ablauf der Mindestverweildauer steht Token allen nachgeschalteten Transitionen zur Verfügung
- Schaltzeitpunkt und -dauer sowie Verzögerungszeit bei Transitionen
  - Während Verz.-Zeit Reservierung der benötigten Tokens oder "Wegschnappen" durch andere Transitionen
  - Dynamisches Setzen von Zeitwerten in Abhängigkeit von Token-Ausprägungen oder der Zeitwerte vorangegangener Transitionen
- Schaltkapazitäten: Wie oft kann eine Transition parallel schalten?
- Relative oder absolute Zeitspezifikationen; Zeitpunkt- oder Intervallbasiert
- Wahrscheinlichkeitsverteilungen bei Zeitpkt. und Intervallen (stochastische Netze)
  - Beispiel: Schaltdauer einer Transition normalverteilt mit Erwartungswert 2h
- Für Workfl. i. allg. relevant: nicht-stoch. Zeitspezifikationen von Transitionen



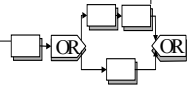
## Zeitorientierte Petri-Netze (2)





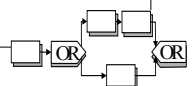
## Objekt-orientierte Petri-Netze

- Anwendung objekt-orientierter Konzepte auf Petri-Netz-Konstrukte
- Objekt-orientierte Tokens
  - Organisation von Token-Typen in Vererbungshierarchien (z.B. Hierarchie von Dokument-Typen)
  - Falls Tokens vom Typ  $T$  in Stelle  $s$  möglich, so auch Tokens vom Typ  $T' < T$  ( $<$  Untertyp-Relation)
  - Tokens als Referenz auf Objekte in OODBMS oder ORDBMS
- Objekt-orientierte Stellen und Transitionen
  - Kapselung von Stellen als abstrakte Datentypen; Schnittstelle zu Datenressourcen
  - Transitionen als abstrakte Datentypen; Schnittstelle zu Applikationen
- Ein Petri-Netz ist ein "Objekt" (mit Methoden etc.)
- Integration mit Middleware-Schichten in verteilten Umgebungen
  - Tokens, Stellen und Transitionen sind miteinander kommunizierende CORBA/DCOM/JAVA-Objekte
  - Petri-Netz spezifiziert Kontroll- und Datenfluss zwischen diesen CORBA/DCOM/JAVA-Objekten

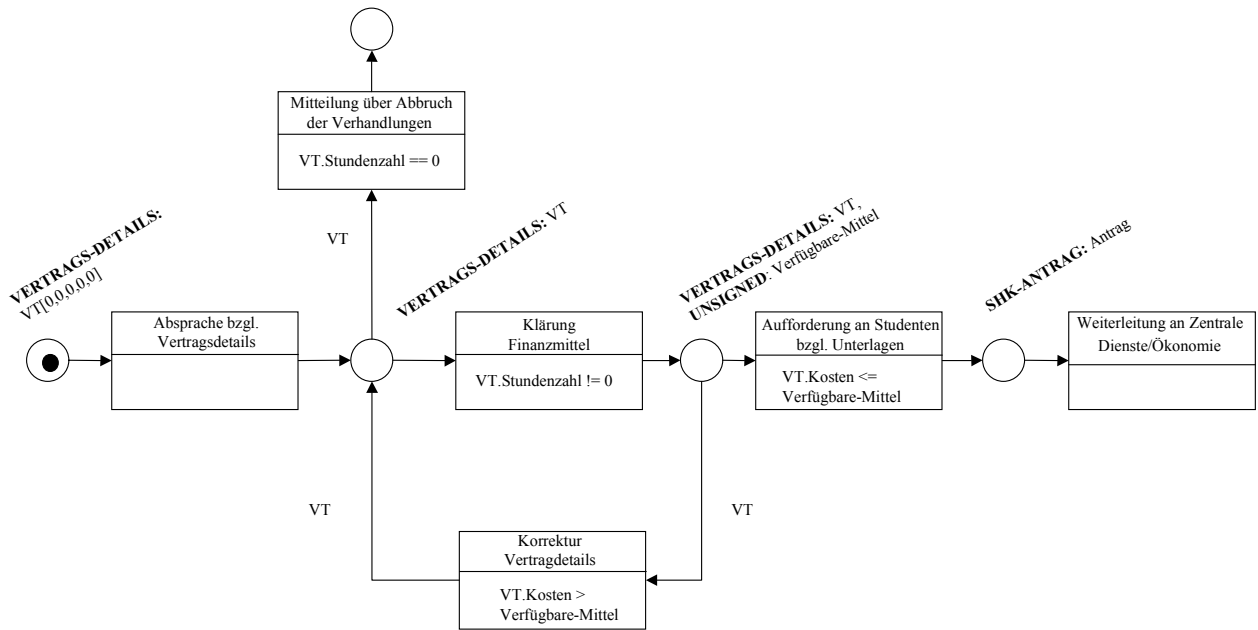


## Fallbeispiel: Token-Typen

Token-Typ	Attribut-Typ	Attribut-Name
<b>VERTRAGS-DETAILS</b>	<b>UNSIGNED</b>	Stundenzahl
	<b>DATE</b>	Beginn
	<b>DATE</b>	Ende
	<b>LIST&lt;STRING&gt;</b>	Tätigkeitsfelder
	<b>UNSIGNED</b>	Kosten
<b>SHK-ANTRAG</b>	<b>VERTRAGS-DETAILS</b>	Details
	<b>DOKUMENT</b>	Lohnsteuerkarte
	<b>DOKUMENT</b>	Lebenslauf
	<b>DOKUMENT</b>	Immatrikulationsbescheinigung
	...	
<b>NACHRICHT</b>	<b>NACHRICHT</b>	Bezug-Auf-Nachricht
	<b>DATE</b>	Gesendet-Am
	<b>STRING</b>	Inhalt
	<b>NACHRICHT</b>	Antwort
	<b>UNSIGNED</b>	Verfügbare-Mittel

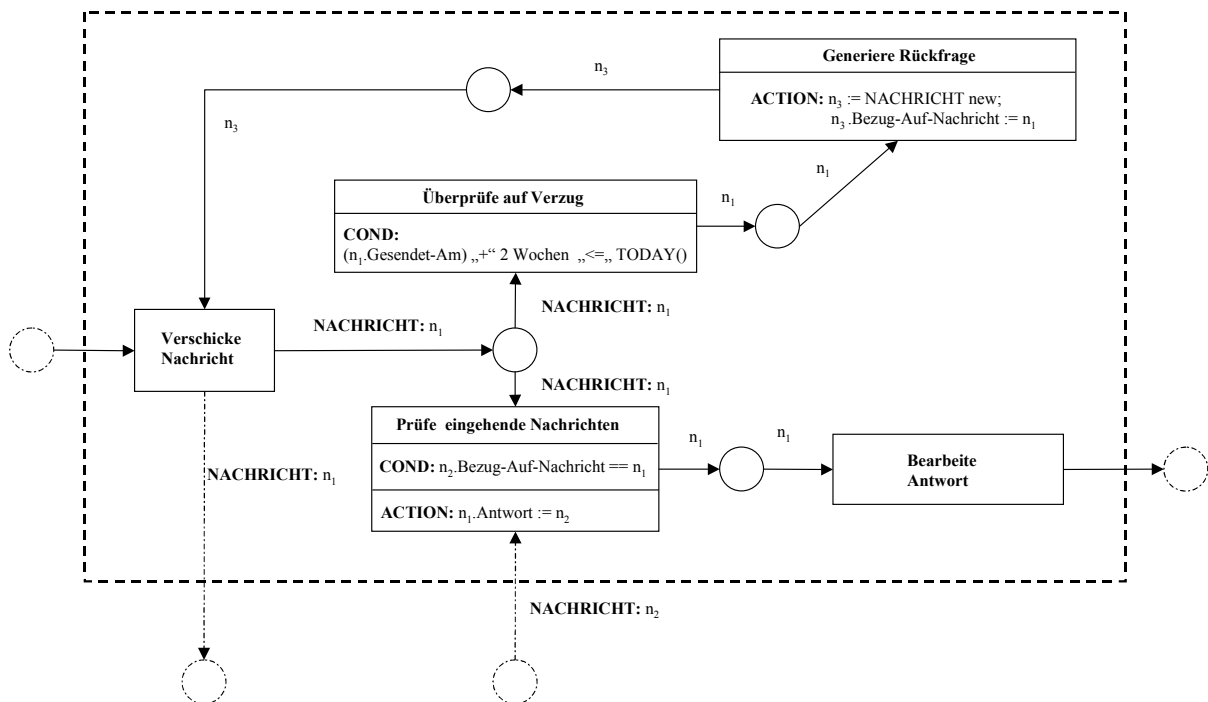


## Fallbeispiel: Petri-Netz für Klärung der Vorbedingungen



## Fallbeispiel: Petri-Netz für Kommunikation mit „Aussenwelt“

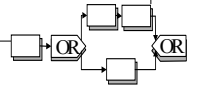
### Abstrakte Transition *Nachricht verschicken*



## Fallbeispiel: Petri-Netz für *Bearbeitung SHK-Antrag*

(C) Prof. E. Rahm, R. Müller

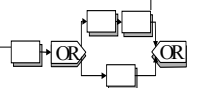
69



## Fallbeispiel: Petri-Netz für *Bearbeitung Hauptvertrag*

(C) Prof. E. Rahm, R. Müller

70



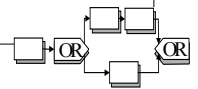
## Petri-Netz-orientiertes Workflow-Management: Produkte und Forschungsgruppen

### ■ Produkte u.a.

- Income (Promatis Software)
- Cosa / Transflow
- IBaan (Baan)

### ■ Forschungsgruppen u.a.

- van der Aalst et al. (Universität Eindhoven; <http://tmitwww.tm.tue.nl/staff/wvdaalst/>)
- Oberweis et al. (Universität Frankfurt; <http://www.wiwi.uni-frankfurt.de/~oberweis/>):  
Petri-Netz-Modellierung mit (NF)<sup>2</sup>-Relationen
- Stefanelli et al. (Universität Pavia): Workflows in der Medizin
- Rahm, Müller et al. (Universität Leipzig; <http://dbs.uni-leipzig.de/de/Research/workflow.html>): Petri-Netz-basiertes Workflow-Management in der verteilten Hämato-Onkologie



## Petri-Netze und Workflows: Zusammenfassung

### ■ Mächtiges formales Modell für die Beschreibung von Kontroll- und Datenflüssen

### ■ Relevante Netztypen u.a.: Pr/T-Netze, FUNSOFT-Netze, zeitorientierte Netze

- Getypte Tokens
- Hierarchische Petri-Netze
- Aktivitätsorientiertes Zugriffs- und Schaltverhalten bzgl. Stellen und Transitionen
- Manuelle und maschinelle Aktivitäten
- Temporale Unterstützung

### ■ Elegante Integration von

- Workflows im engeren Sinn (z.B. *eine Menge von Personen und Applikationen bearbeitet ein Dokument oder Werkstück bis zur Fertigstellung*) und
- Datenverteilungs- und Kommunikationsprozessen (*welches Dokument oder welche Nachricht ist wo im System mit welchem Bezug eingegangen*)

### ■ Datenzustand eines Petri-Netzes explizit (Datenzustand = Netz-Markierung)

### ■ „Nachteile“:

- Kontrollfluss wird „gesteuert“ vom Datenfluss (d.h. Tokenfluss)
- Für Anwender schwer zu verstehen (vor allem wegen Dualismus Stelle ↔ Transition)