

# ***Modul OMSI-2*** ***im SoSe 2010***

## ***Objektorientierte Simulation*** ***mit ODEMx***

Prof. Dr. Joachim Fischer  
Dr. Klaus Ahrens  
Dipl.-Inf. Ingmar Eveslage  
Dipl.-Inf. Andreas Blunk

[fischer|ahrens|eveslage|blunk@informatik.hu-berlin.de](mailto:fischer|ahrens|eveslage|blunk@informatik.hu-berlin.de)

## 9. *SDL-Konzepte* (Präzisierung, 2. Teil)

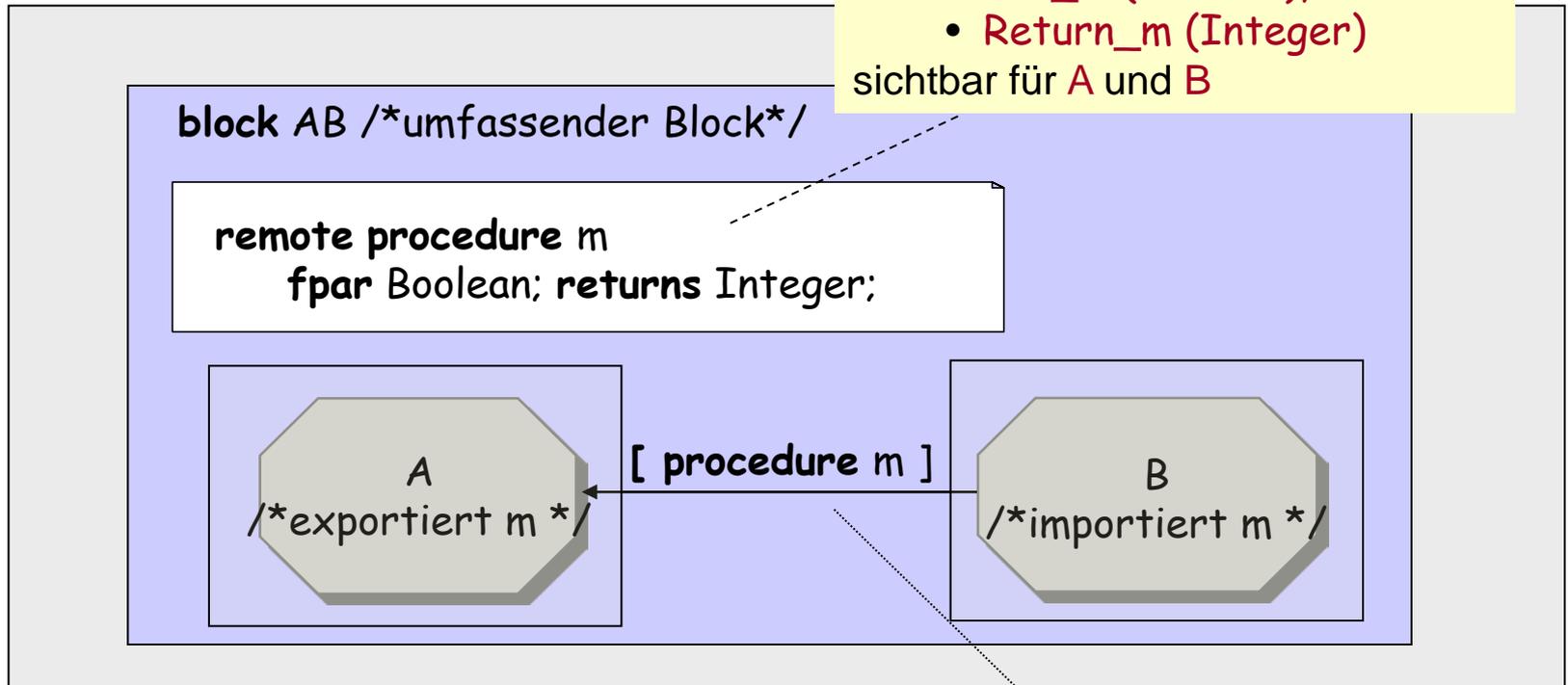
1. Ersetzungskonzept: Shorthand-Notation
2. Ersetzungskonzept: Priorisierter Input
3. Prozeduren, Ersetzungskonzept: Remote-Prozeduren
4. Programm-Ausnahmen (SDL-2000-Konzept)
5. Spezialisierung von Zustandsautomaten
6. Lokale Objekte, Semaphore

# Deklaration: Remote-Prozeduren (Wdh.)

Transformation in implizite Signale:

- **Call\_m** (Boolean),
- **Return\_m** (Integer)

sichtbar für A und B



Namen von Remote-Prozeduren können in

- Signallisten,
- Kanälen, Routen, Gates und
- Signalsets

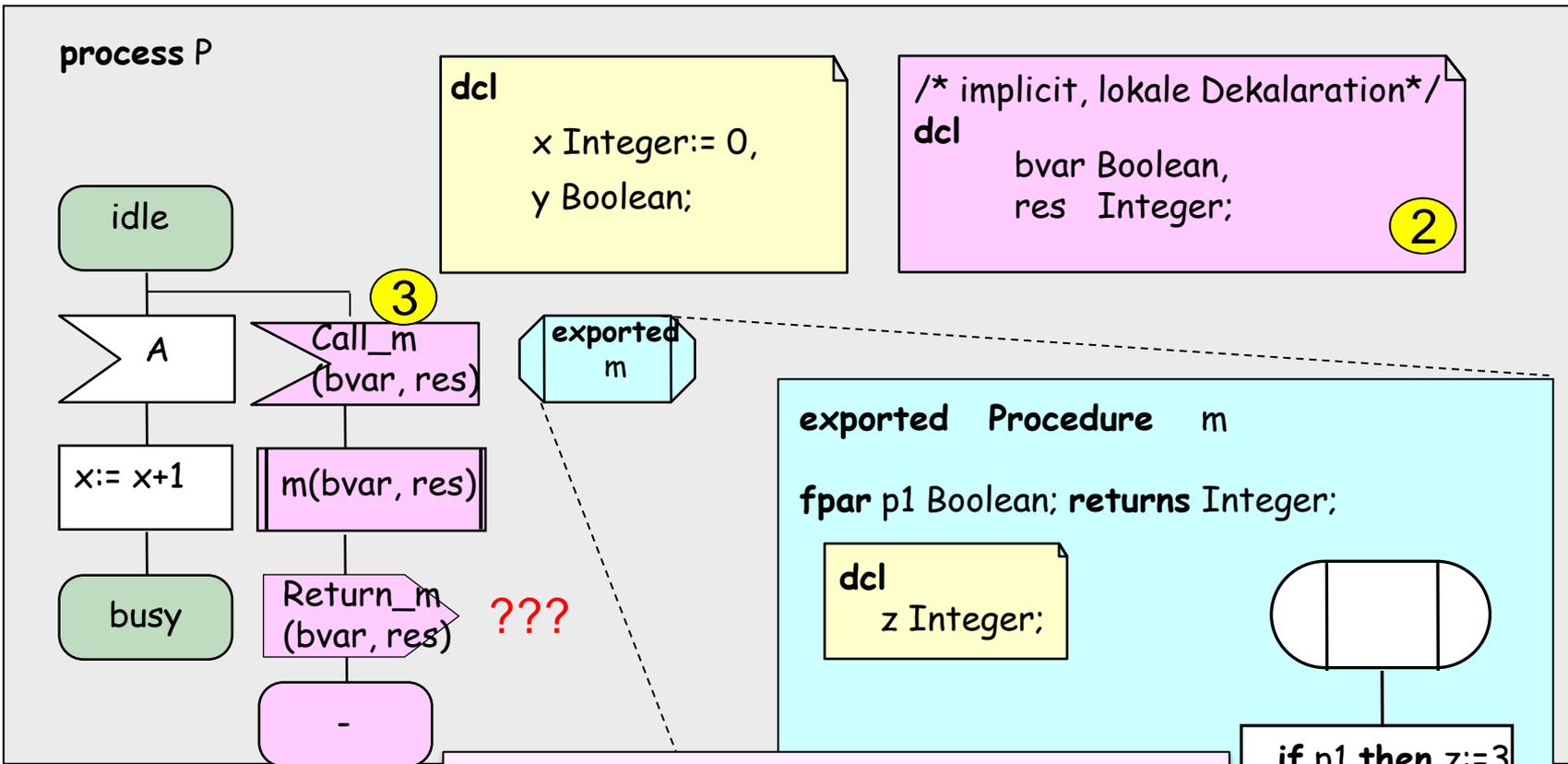
erscheinen (wenn nicht, dann implizit)

Angabe aus Richtung  
des Rufers/Importeurs  
(obwohl bi-direktional)

# Idee für Exporteur einer Meta

1

```
/* implicit, globale Deklaration */
signal
  Call_m (Boolean, Integer),
  Return_m (Boolean, Integer)
```



```
dcl
  x Integer := 0,
  y Boolean;
```

```
/* implicit, lokale Deklaration */
dcl
  bvar Boolean,
  res Integer;
```

```
exported Procedure m
  fpar p1 Boolean; returns Integer;
  dcl
    z Integer;
```

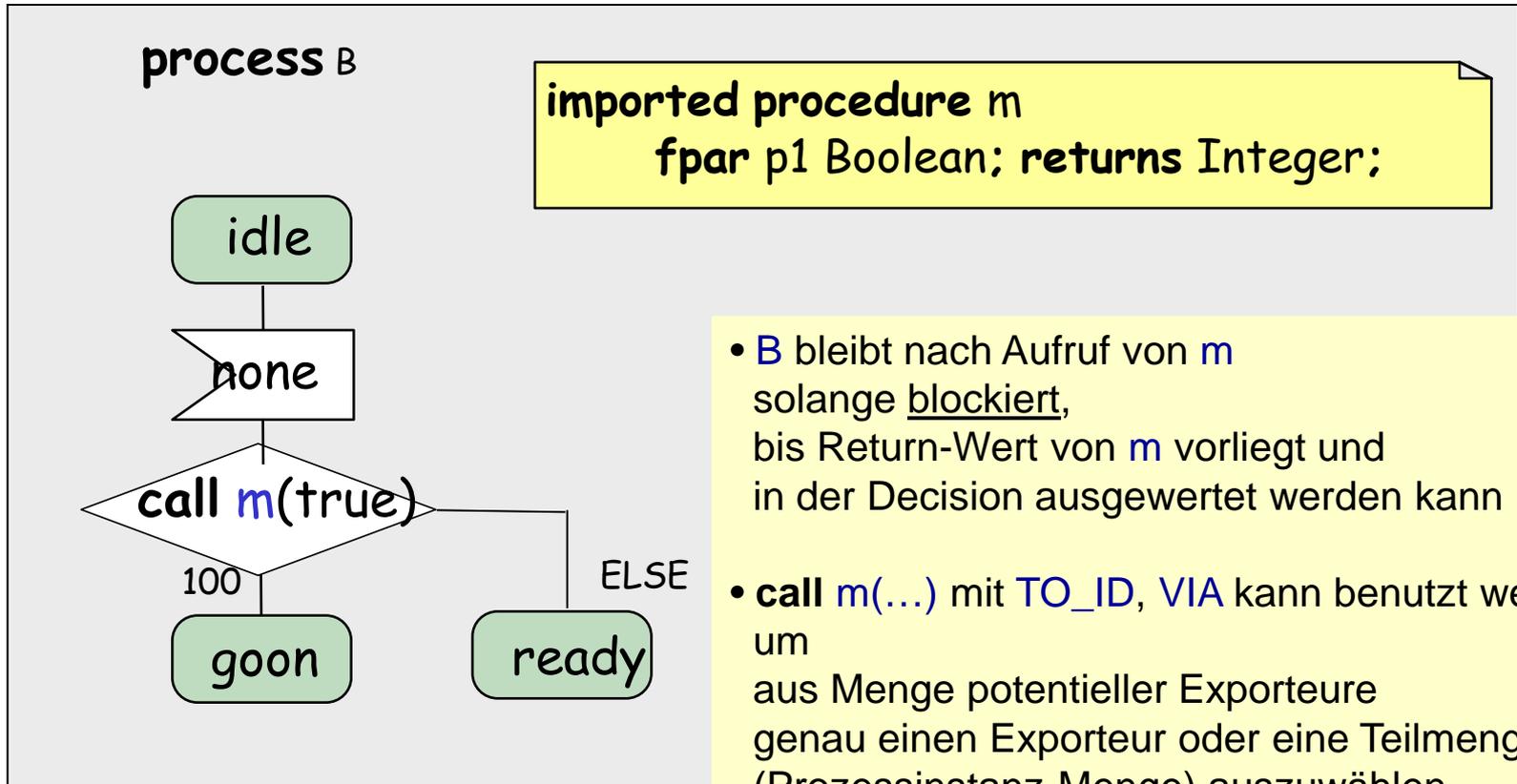
In **jedem** Zustand von **P** werden **zusätzlich** implizite Trigger eingeführt:

- Input für Call-Requests für **m** (**Call\_m**)
- Aktionen: Aufruf von **m** mit Resultatübernahme
- Output von **Return\_m** mit Resultat
- Zustandswechsel

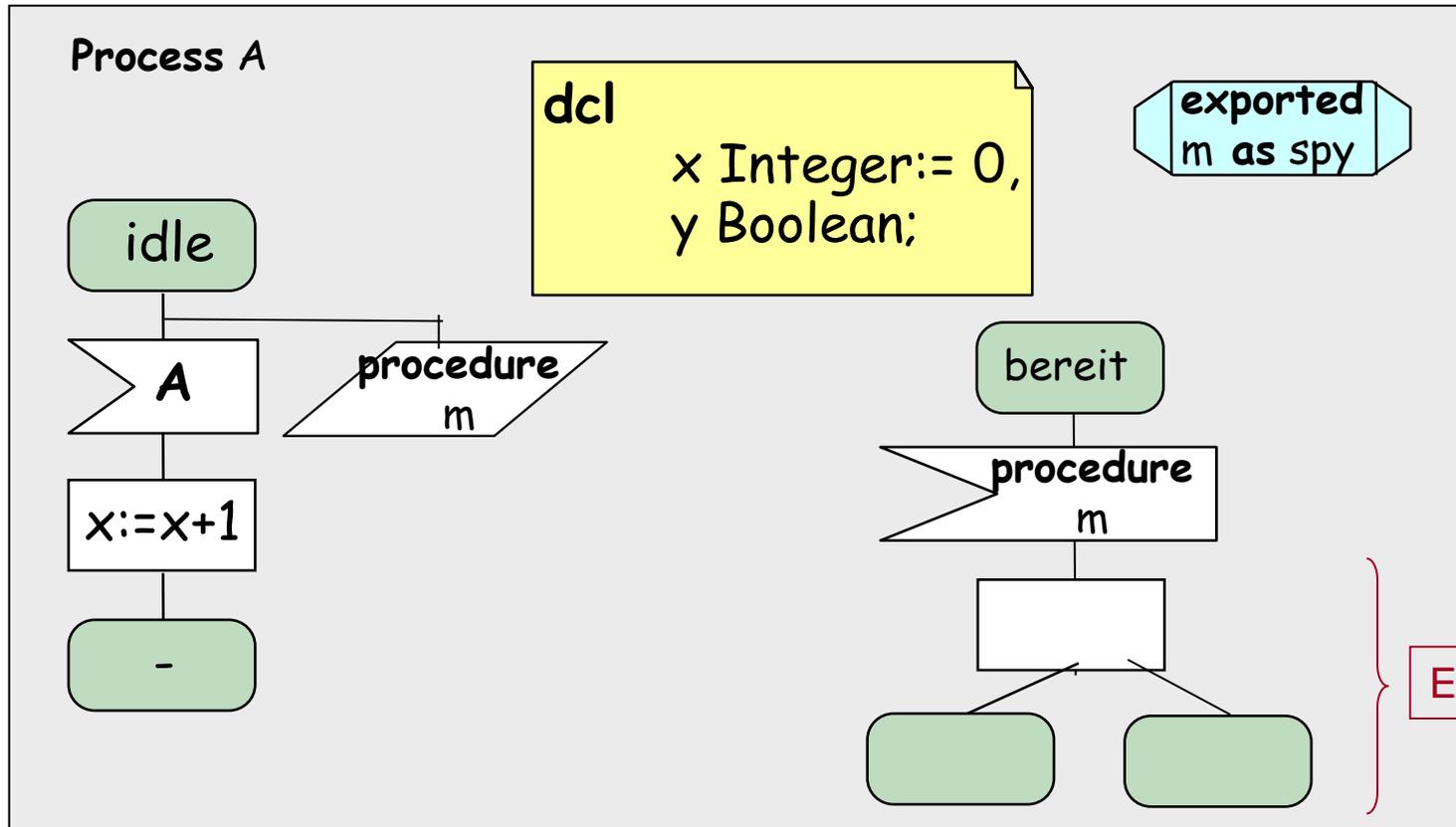
Rangfolge:  
FCFS, wie normale Input-Signal-Trigger

3

# Idee für Importeur einer Methode (Wdh.)



# Idee für Verhaltensflexibilität des Exporteurs (Wdh.)



## Erweiterung der Standardsemantik:

1. Prozedur kann unter anderem Namen exportiert werden (hier: **spy**)
2. explizite Angabe von Zuständen, die Call-Requests akzeptieren (hier: **bereit**) bei zusätzlicher Zustandsgraph-Erweiterung
3. explizite Angabe von Zuständen, die Call-Requests zurückstellen (hier: **idle**)

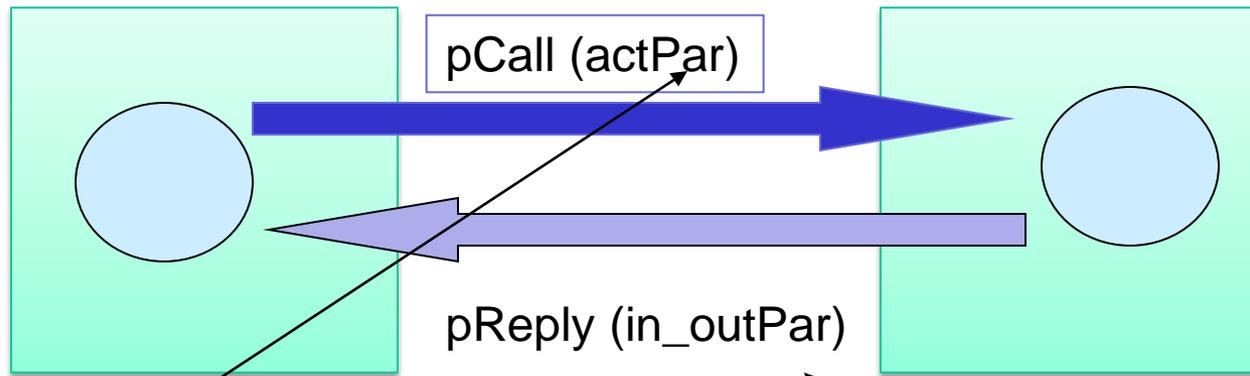
# RPC-Ersetzungsmodell

Remote procedure  $p (...)$

- 1 Aus RPC-Deklaration  $\rightarrow$  implizite Signaldefinition + implizite bidirektionaler Kanal

Client

Server



Liste aktueller In/Out-Parameter

Liste aktueller InOut-Parameter (inkl. Return-Wert)

# RPC-Ersetzungsmodell des Client

**CALL** p (actPar, in\_outParVar) **TO\_ID** destination;

3



**OUTPUT** p**CALL**(actPar, in\_outParVar) **TO\_ID** destination;  
warten im Zustand **pWait** bei Rettung aller Signale mit  
**INPUT** p**REPLY**(in\_outParVar);

2

implizite Deklaration von Zustand  
entsprechender lokaler Variablen

4

in allen anderen Zuständen wird **pREPLY** gerettet

# RPC-Ersetzungsmodell des Client

bei allen Zuständen des Servers mit  
**INPUT procedure p**  
<Transition>

6

wird wird folg. Input-Teil hinzugefügt

```
INPUT pCALL(actParVar);  
TASK senderVar:= SENDER;  
CALL local_p (actParVar, in_outParVar);  
OUTPUT pREPLY(in_outParVar) TO_ID senderVar;  
<Transition>
```

5

bei impliziter Deklaration  
entsprechender Variablen

7

bei allen Zuständen des Servers mit

**SAVE procedure p**  
erfolgt eine Ersetzung mit **SAVE pCALL;**

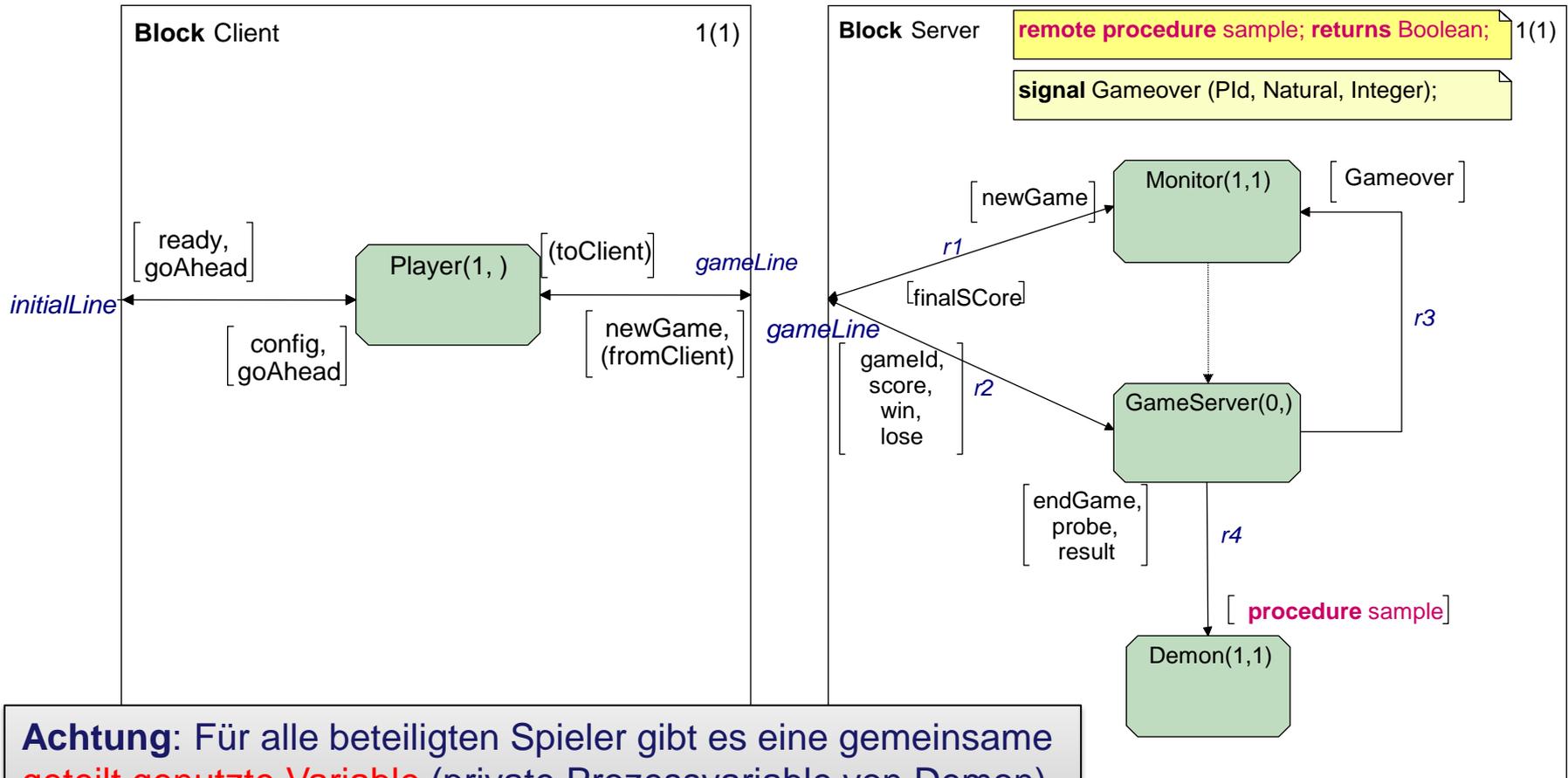
# RPC-Ersetzungsmodell des Client

bei allen anderen Zuständen des Servers  
(mit Ausnahme der impliziten Zustände)  
wird folg. Input-Teil ergänzt:

8

```
INPUT pCALL(actParVar);  
TASK senderVar:= SENDER;  
CALL local_p (actParVar, in_outParVar);  
OUTPUT pREPLY(in_outParVar) TO_ID senderVar;  
NEXTSTATE -;
```

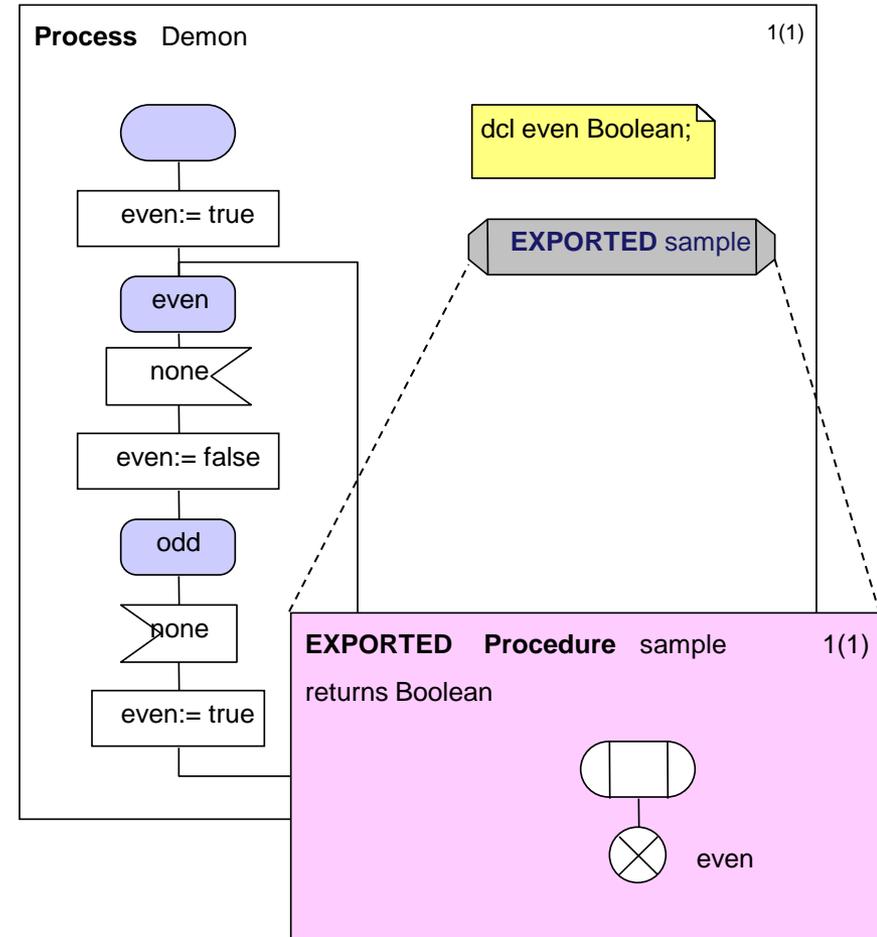
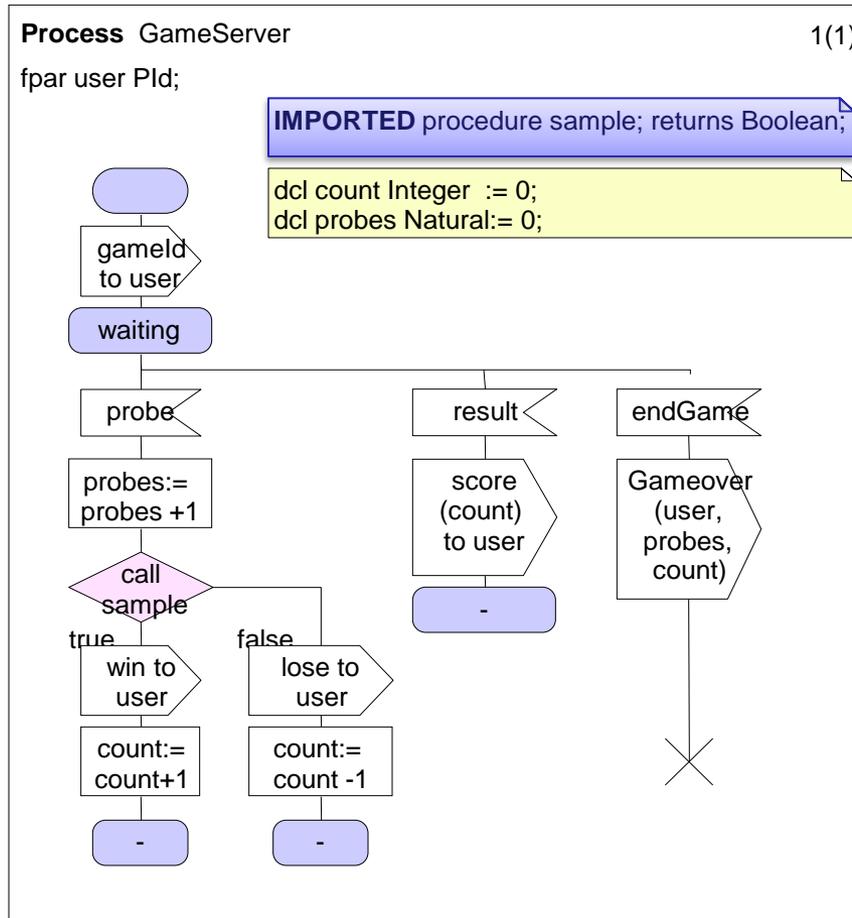
# RPC-Beispiel: Demon-Game in Standard-SDL



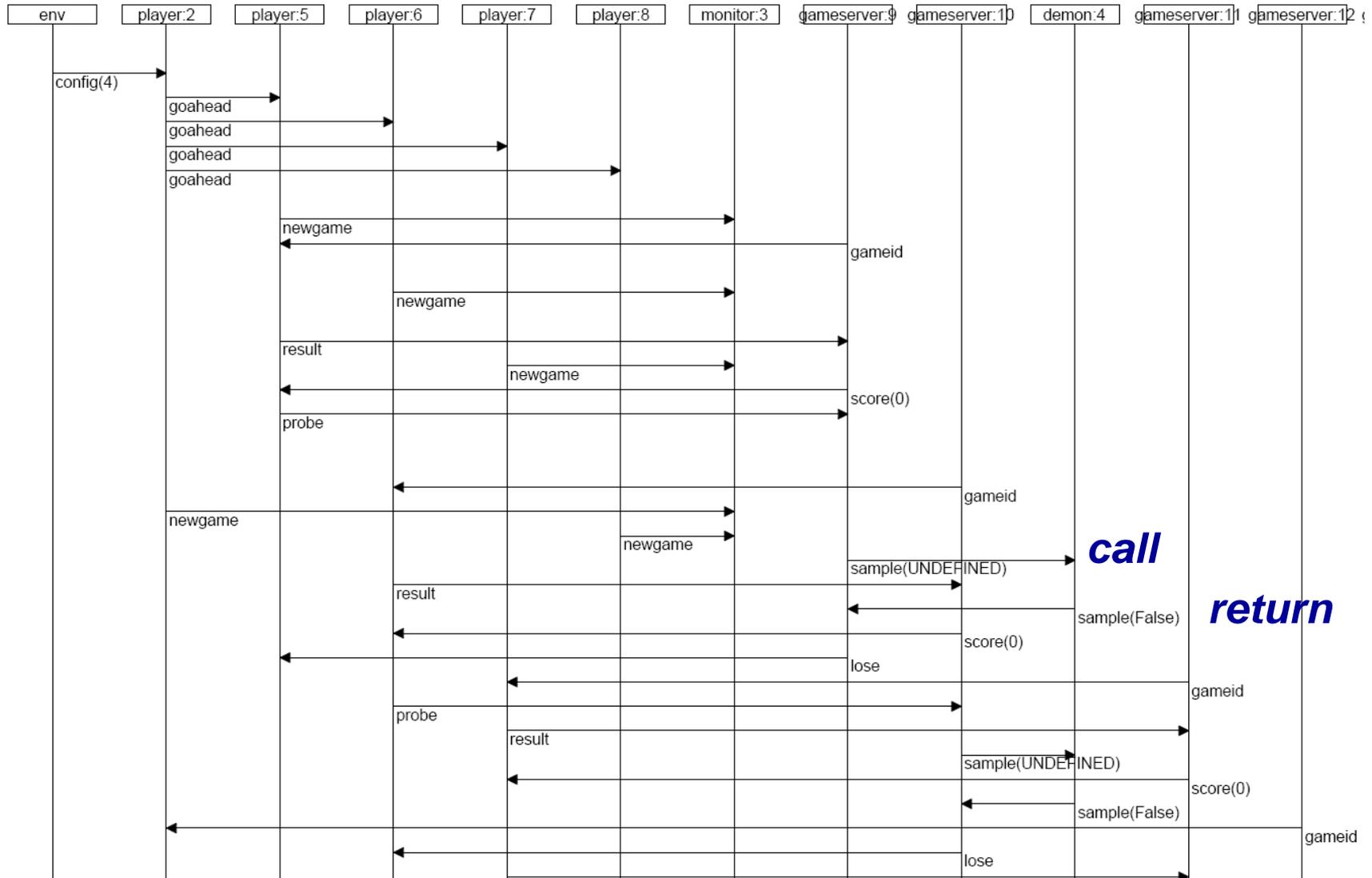
**Achtung:** Für alle beteiligten Spieler gibt es eine gemeinsame **geteilt genutzte Variable** (private Prozessvariable von Demon)  
 Ein gleichzeitiger Zugriff per **Remote Prozedur** muss zwangssequentialisiert werden

# Dreiklang: Remote – Imported - Exported

REMOTE procedure sample; returns Boolean;



# Ablauf mit Ruf und Return einer Remote-Prozedur



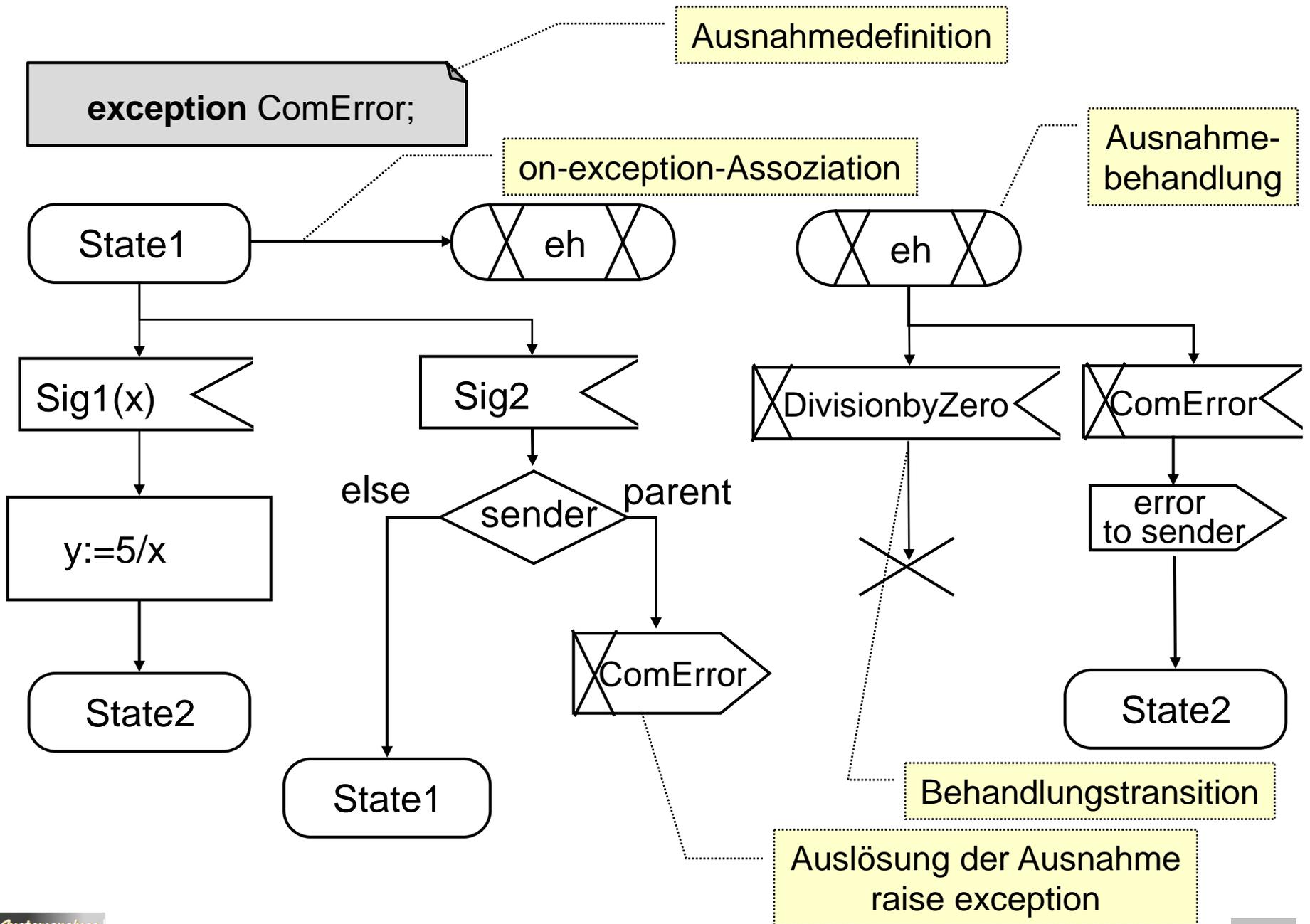
## 9. *SDL-Konzepte* (Präzisierung, 2. Teil)

1. Ersetzungskonzept: Shorthand-Notation
2. Ersetzungskonzept: Priorisierter Input
3. Prozeduren, Ersetzungskonzept: Remote-Prozeduren
4. Programm-Ausnahmen (SDL-2000-Konzept)
5. Spezialisierung von Zustandsautomaten
6. Lokale Objekte, Semaphore

# Erweiterte Zustandsmaschinen

**Ausnahmen** werden verwendet, um unerwartetes oder unnormales Verhalten auszudrücken:

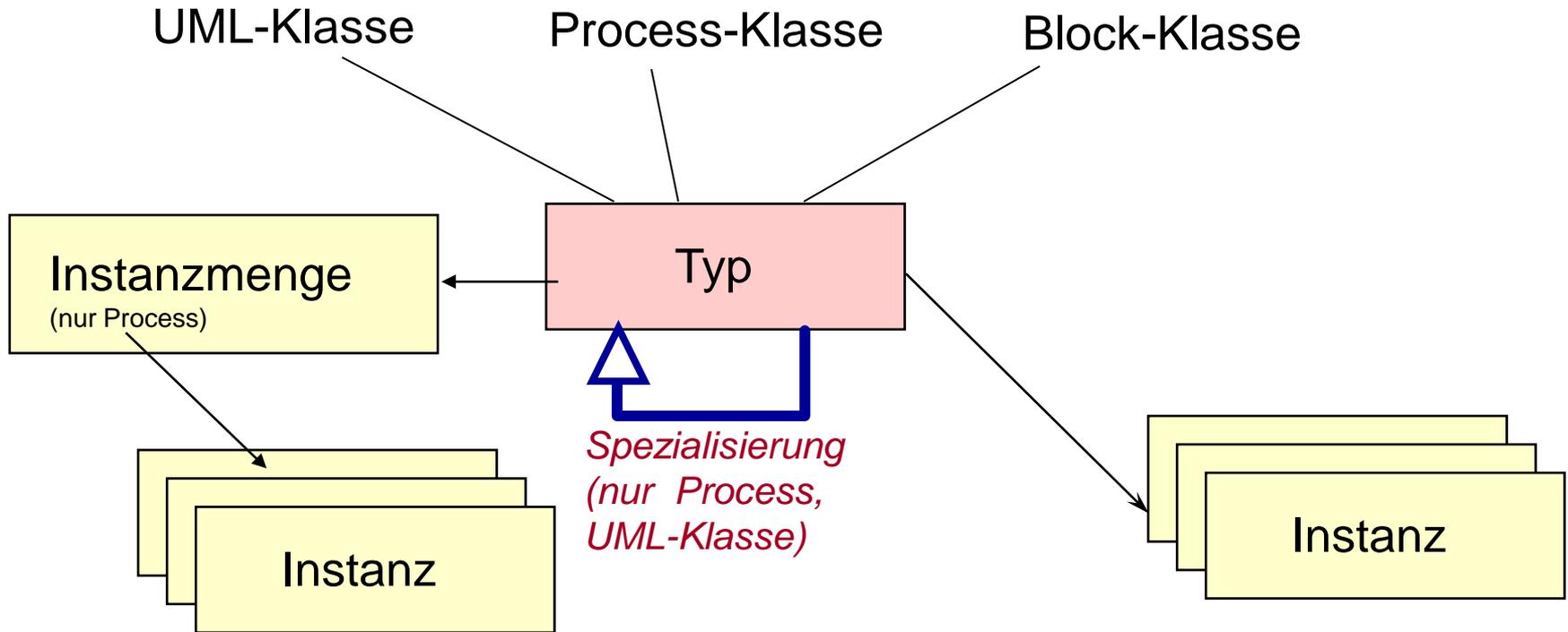
- **exception**: Typ der Ausnahme
- **exception handler**:  
Verhalten, das bei Auftreten einer Ausnahme ausgeführt werden soll
- **onexception**:  
Zuweisung eines *exception handler* an Einheit der Verhaltensbeschreibung eines Agenten
- **raise**: Löst explizit eine Ausnahme in einer Transition aus  
(es gibt Standardausnahmen, die implizit ausgelöst werden  
z.B. **DivisionbyZero**, ...)



## 9. *SDL-Konzepte* (Präzisierung, 2. Teil)

1. Ersetzungskonzept: Shorthand-Notation
2. Ersetzungskonzept: Priorisierter Input
3. Prozeduren, Ersetzungskonzept: Remote-Prozeduren
4. Programm-Ausnahmen (SDL-2000-Konzept)
5. Spezialisierung von Zustandsautomaten
6. Lokale Objekte, Semaphore

# Typen, parametrisierte Typen, Spezialisierung

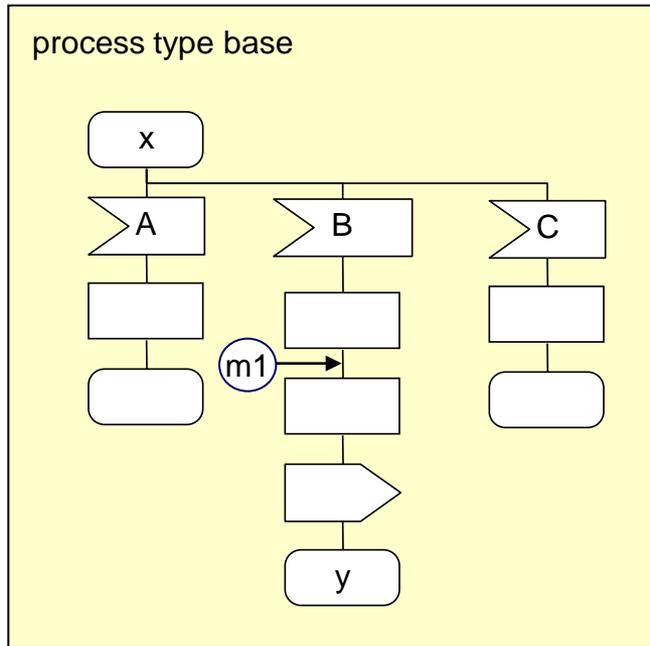


Vererbung in SDL/RT stark eingeschränkt:  
Process Typ und UML-Klasse

# Spezialisierung (Vererbung)

- Mechanismus, um einen neuen Typ von einem bereits existenten Typ (Basistyp) abzuleiten
- dabei können Konzepte des Basistyps
  - übernommen (geerbt)
  - modifiziert und
  - erweitert werden
- der Basistyp legt fest, was modifiziert werden kann und was nicht
  - **virtual** kennzeichnet Urdefinition und kann redefiniert werden
  - **redefined** kennzeichnet Redefinition und kann **erneut** redefiniert werden
  - **finalized** kennzeichnet Redefinition und kann **nicht mehr** redefiniert werden

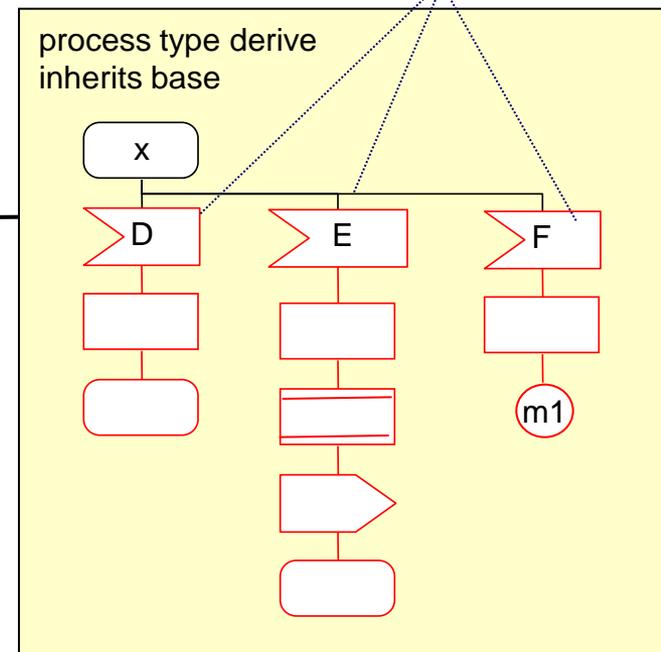
# Process-Typ-Spezialisierung: Erweiterung des Zustandsgraphen



**Achtung:**

In SDL haben Basis und Ableitung einen gemeinsamen Namensraum !!!

*zusätzliche Übergänge für x*

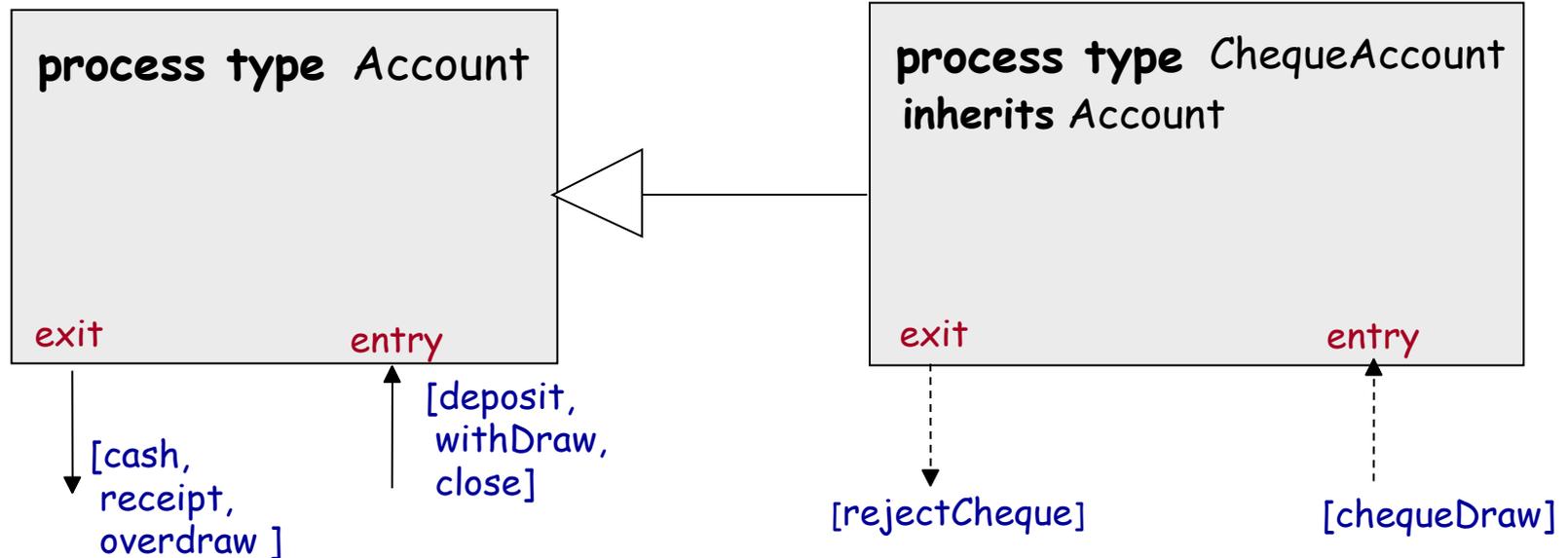


Die Automaten-Erweiterung um eine Transition ist nur möglich, wenn die Transition (Zustand, Trigger) **nicht** bereits im Basis-Zustandsgraph vorkommt

# Process-Typ-Spezialisierung: Vererbungsbeispiel

Basistyp

Spezialisierung/ abgeleiteter Typ  
(mit Gate-Erweiterung)



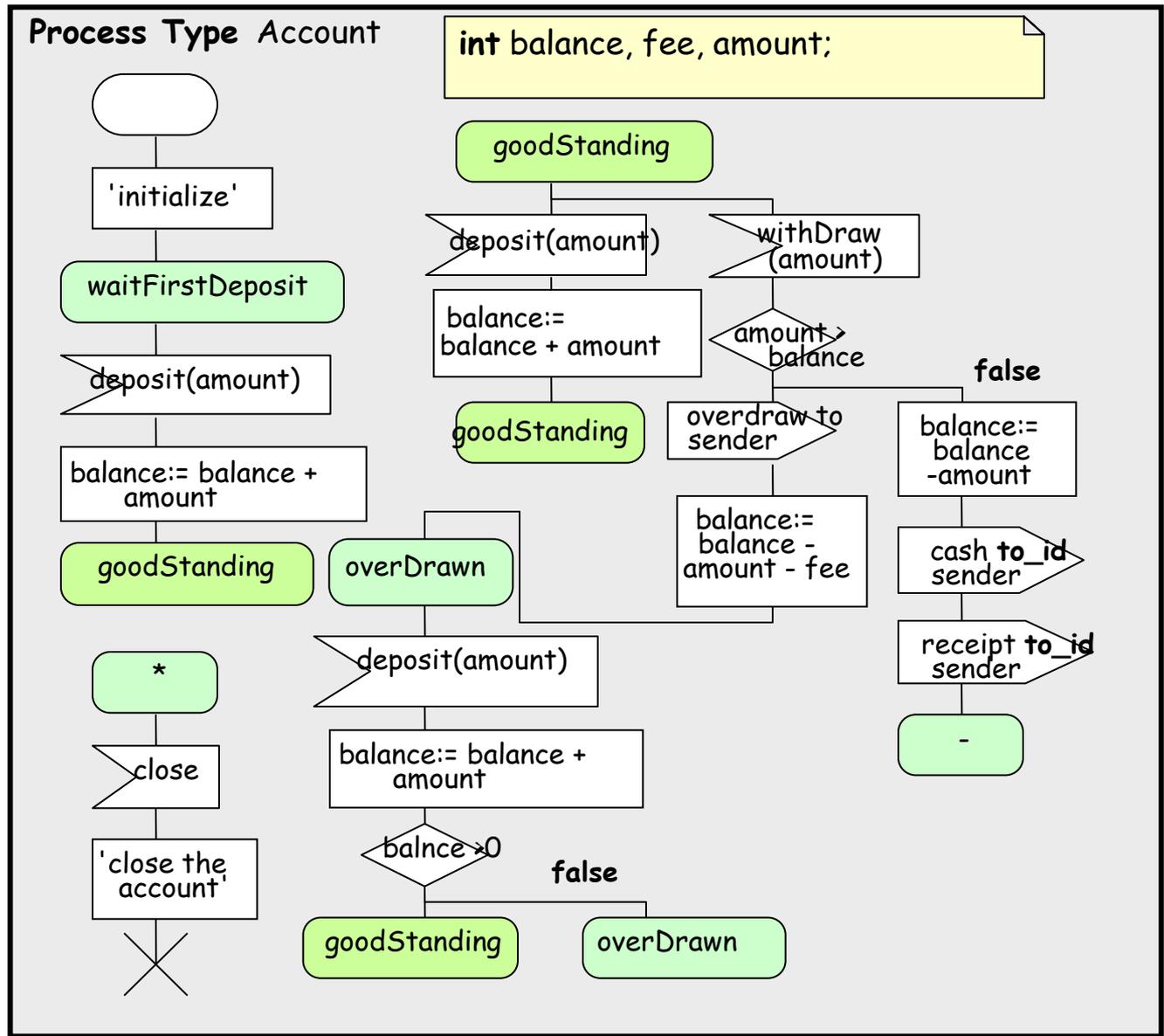
## Account- Funktionalität

- Einzahlen
- Abheben → (Cash, Quittung) |  
(Überziehungsmitteilung)
- Konto schließen

## zusätzliche ChequeAccount- Funktionalität

- Überweisung → (Annullierung)

# Basistyp



exit

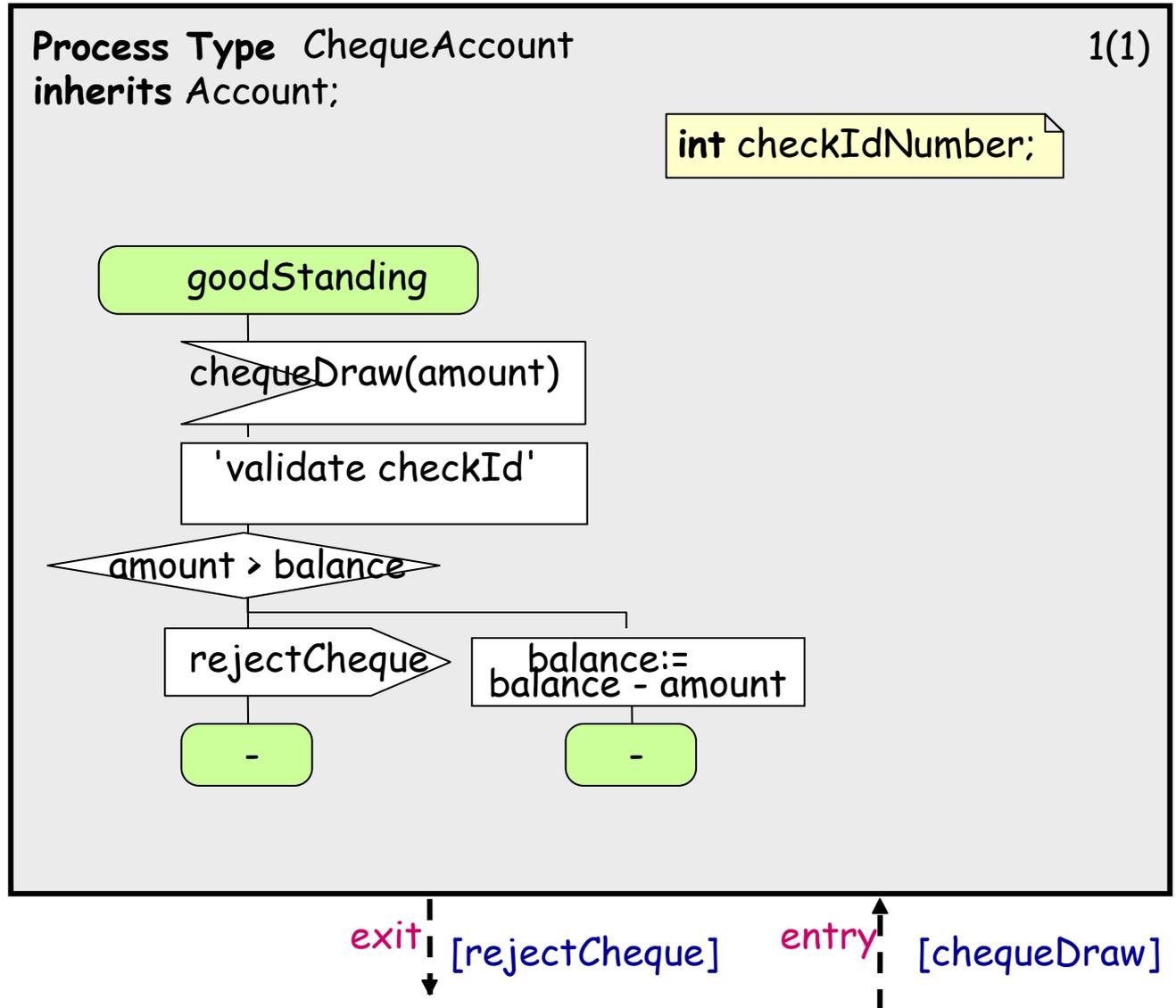
[cash, receipt, overdraw]

OMSI-2 - SDL-Einführung

[deposit, withDraw, close]

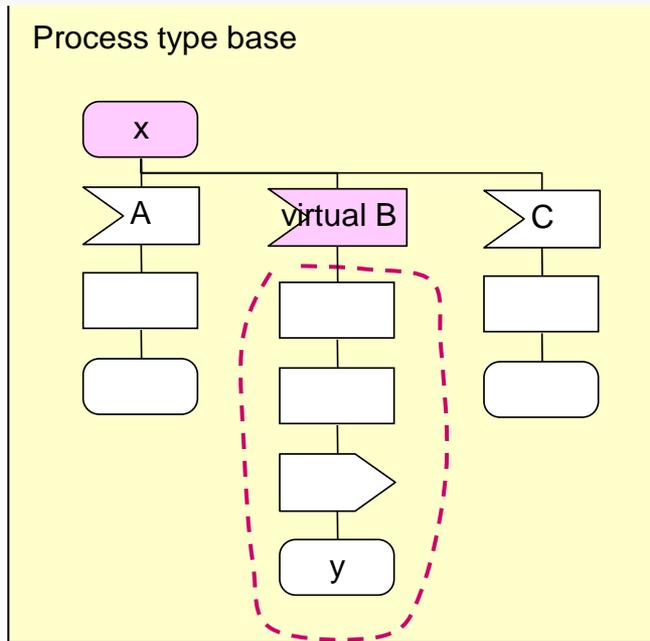
entry

# Ableitung



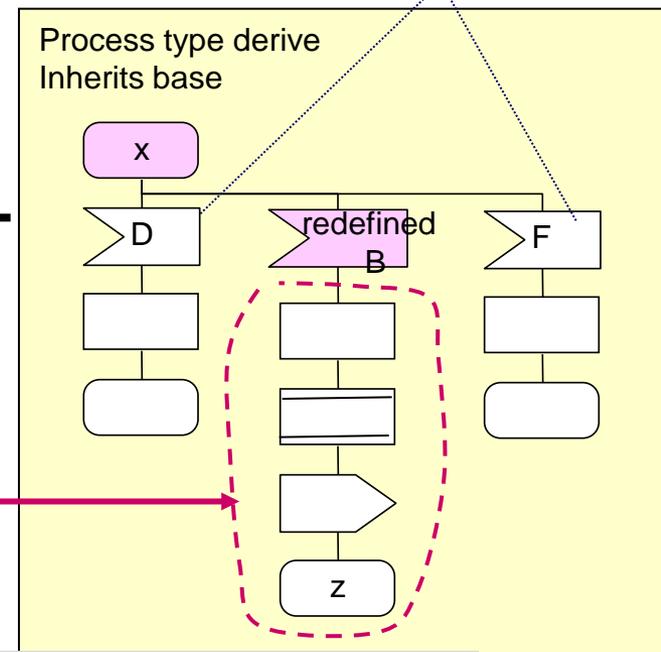
# Redefinition einzelner Transitionen: Prinzip

gilt für Process-Typ, Service-Typ, Procedure

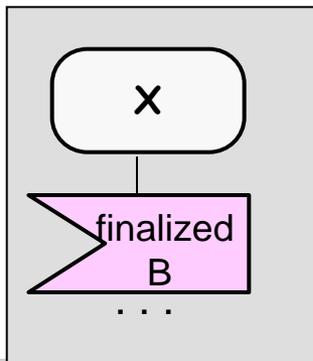


**eindeutige Bezugnahme:**  
Zustand, virtueller Auslöser

*zusätzliche Übergänge*

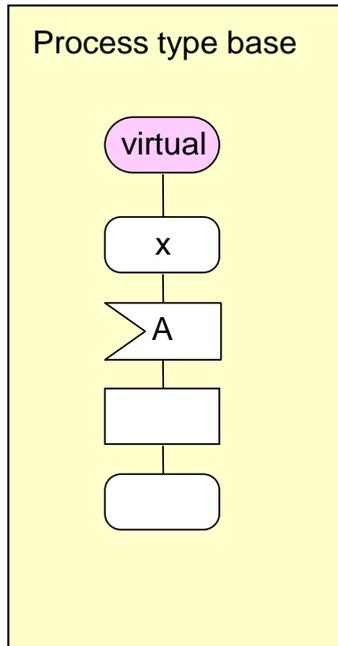


*wird ersetzt*

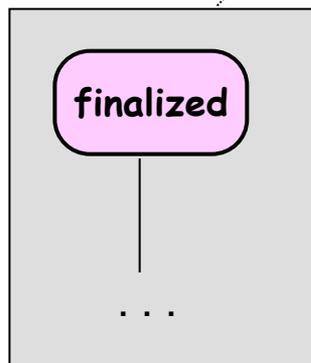
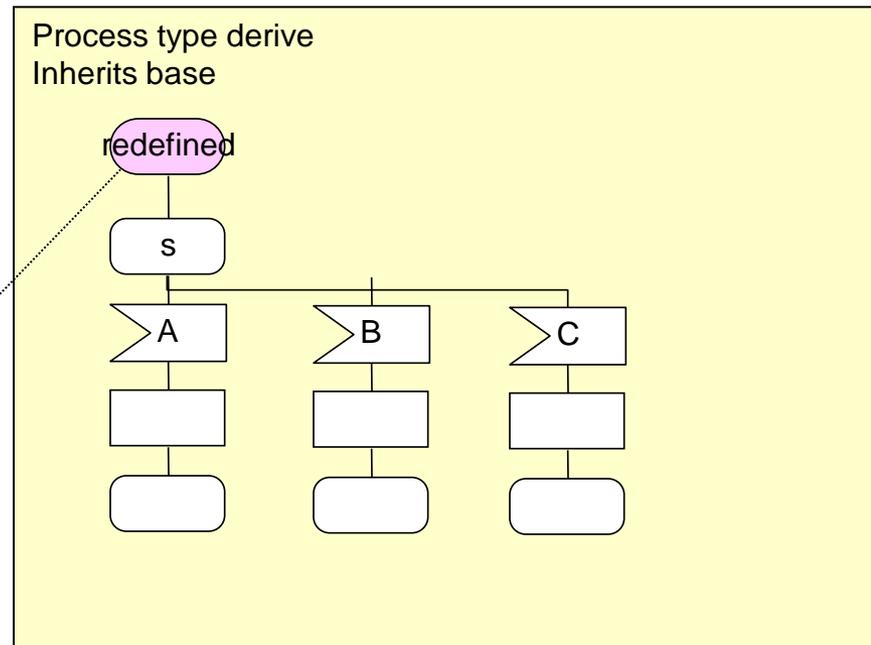


**Achtung:** lokale Variablen  
zur Übernahme der Nachrichten-Parameter änderbar

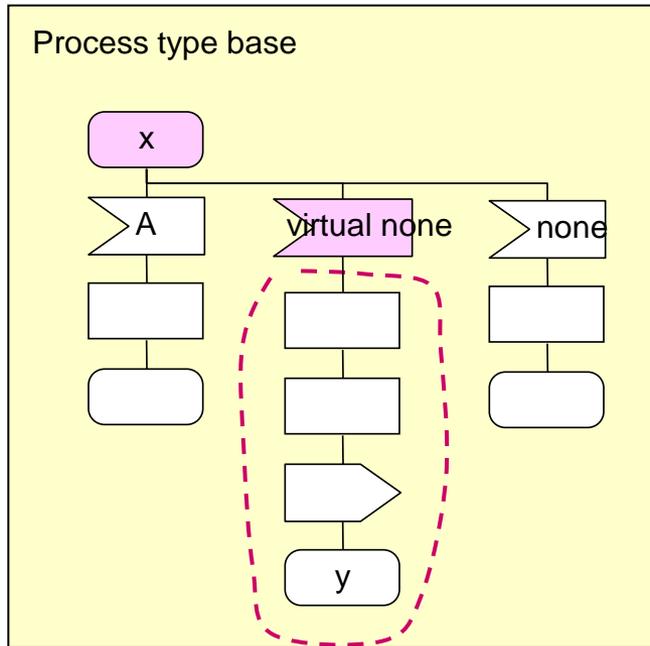
# Redefinition des gesamten Zustandsautomaten



**eindeutige Bezugnahme:**  
nur ein Startzustand erlaubt



# Redefinition spontaner Transitionen

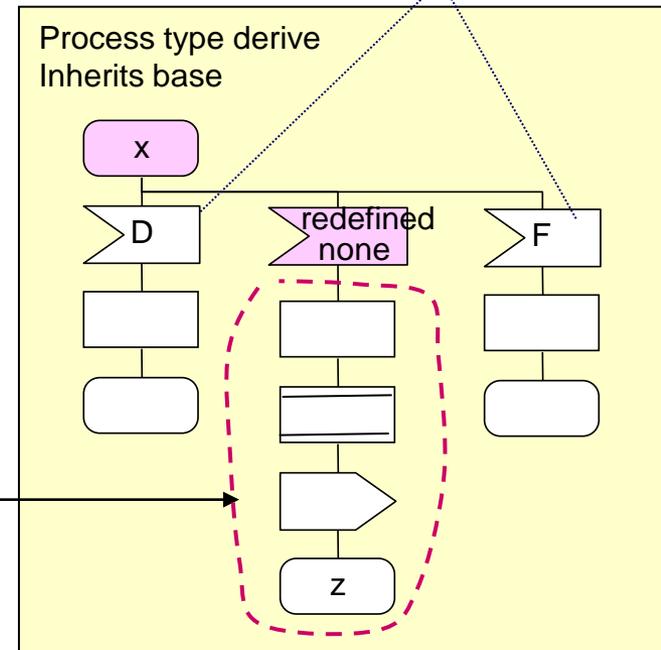


wird ersetzt

eindeutige Auflösung  
macht Einschränkung erforderlich:

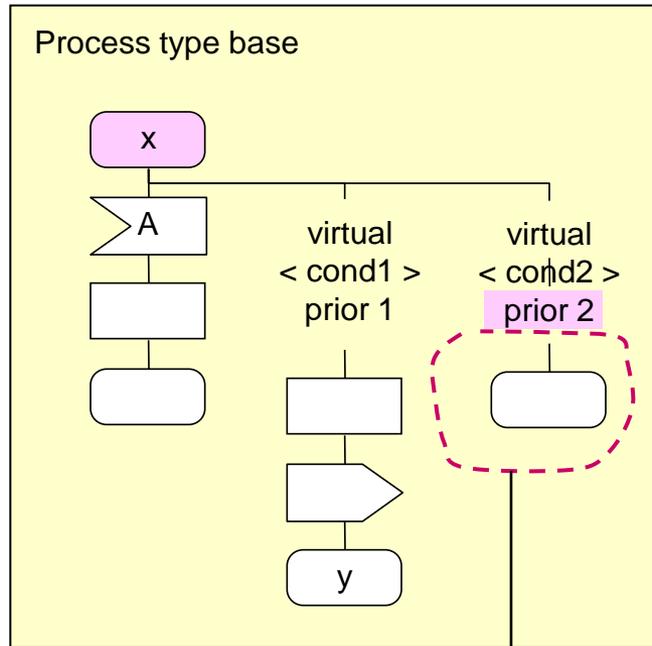
→ pro Zustand ist im Basistyp **nur eine**  
virtuelle spontane Transition erlaubt

zusätzliche Übergänge



nicht in SDL/RT !

# Redefinition von Continuous-Transitionen

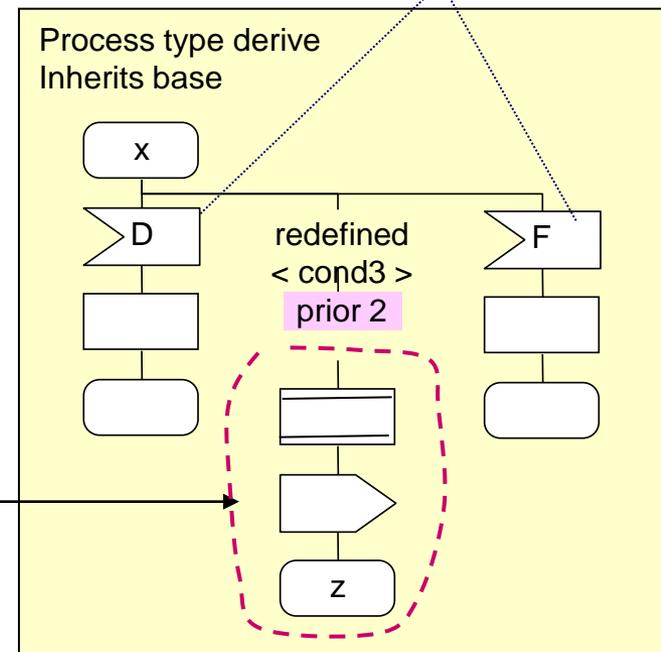


**eindeutige Auflösung**

macht Einschränkung erforderlich:

→ pro Zustand müssen sich im Basistyp die virtuellen Continuous-Transitionen in der **Priorität** unterscheiden

*zusätzliche Übergänge*



*wird ersetzt,  
sogar die Bedingung  
kann sich ändern !*

# Flexible Redefinition von Transitionen mit Signal-Triggern

- ein virtueller Trigger für einen Zustand S eines Basis-Process-Typs X der folgenden Art
  - input virtual sig
  - save virtual sig
  - ~~priority input virtual sig~~
  - ~~input virtual sig provided <expr 1>~~
- kann in einer Ableitung von X im Zustand S zu einem beliebigen Trigger der folgenden Art umfunktioniert werden
  - input redefined sig
  - save redefined sig
  - ~~priority input redefined sig~~
  - ~~input redefined sig provided <expr 2>~~

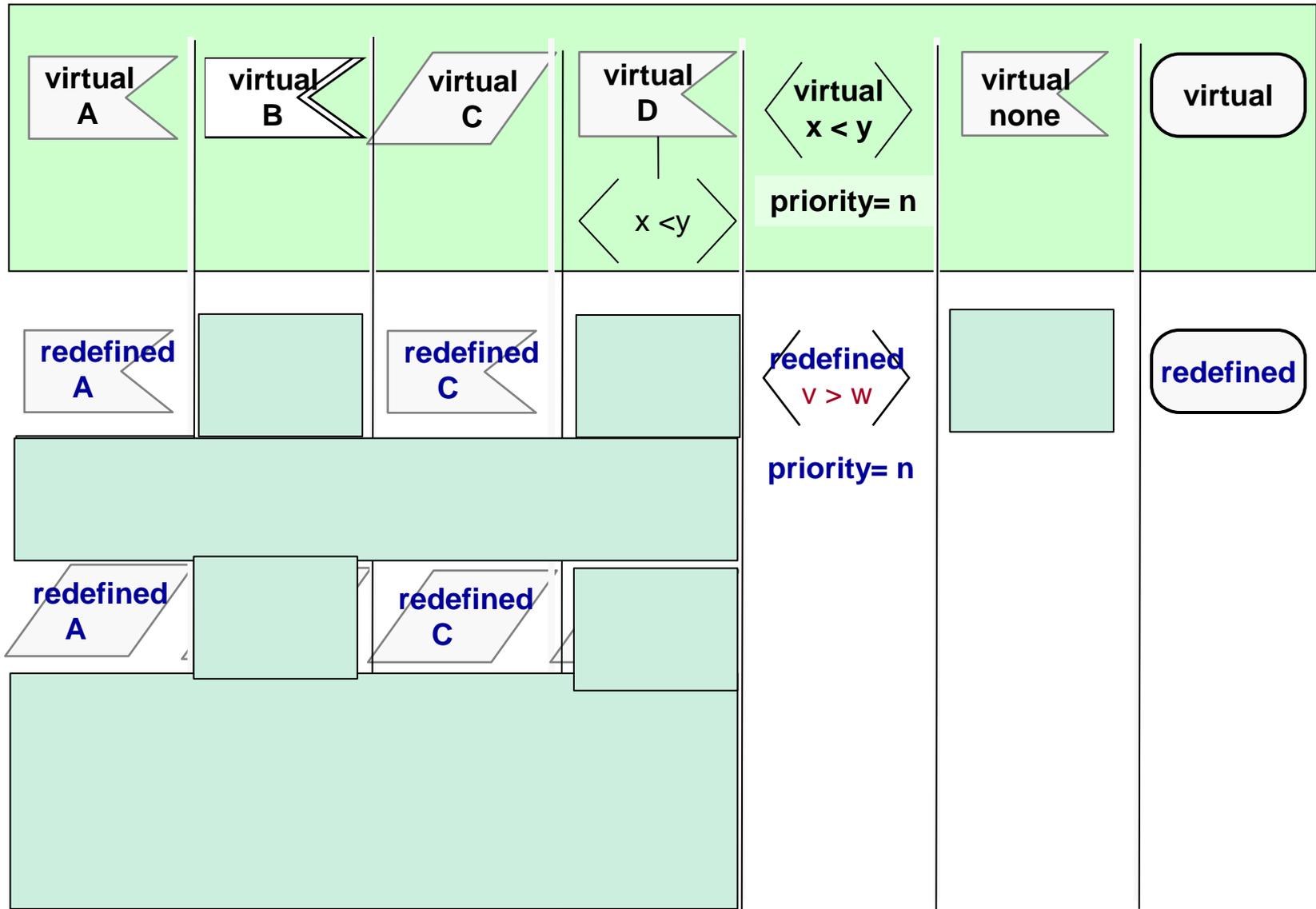
(Zustandsname,  
Signalname)

1-deutige  
Zuordnung

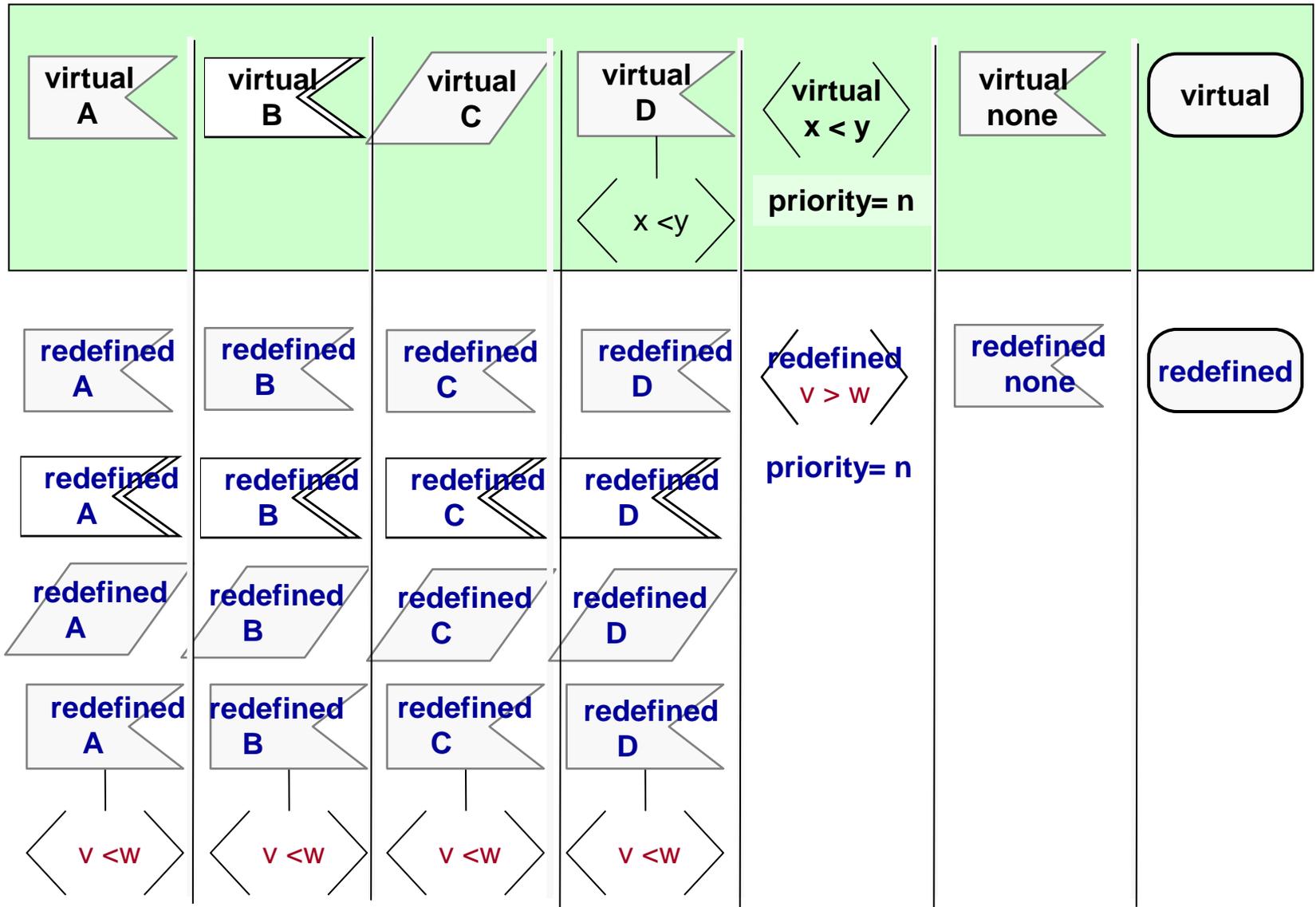
(Zustandsname,  
Signalname)



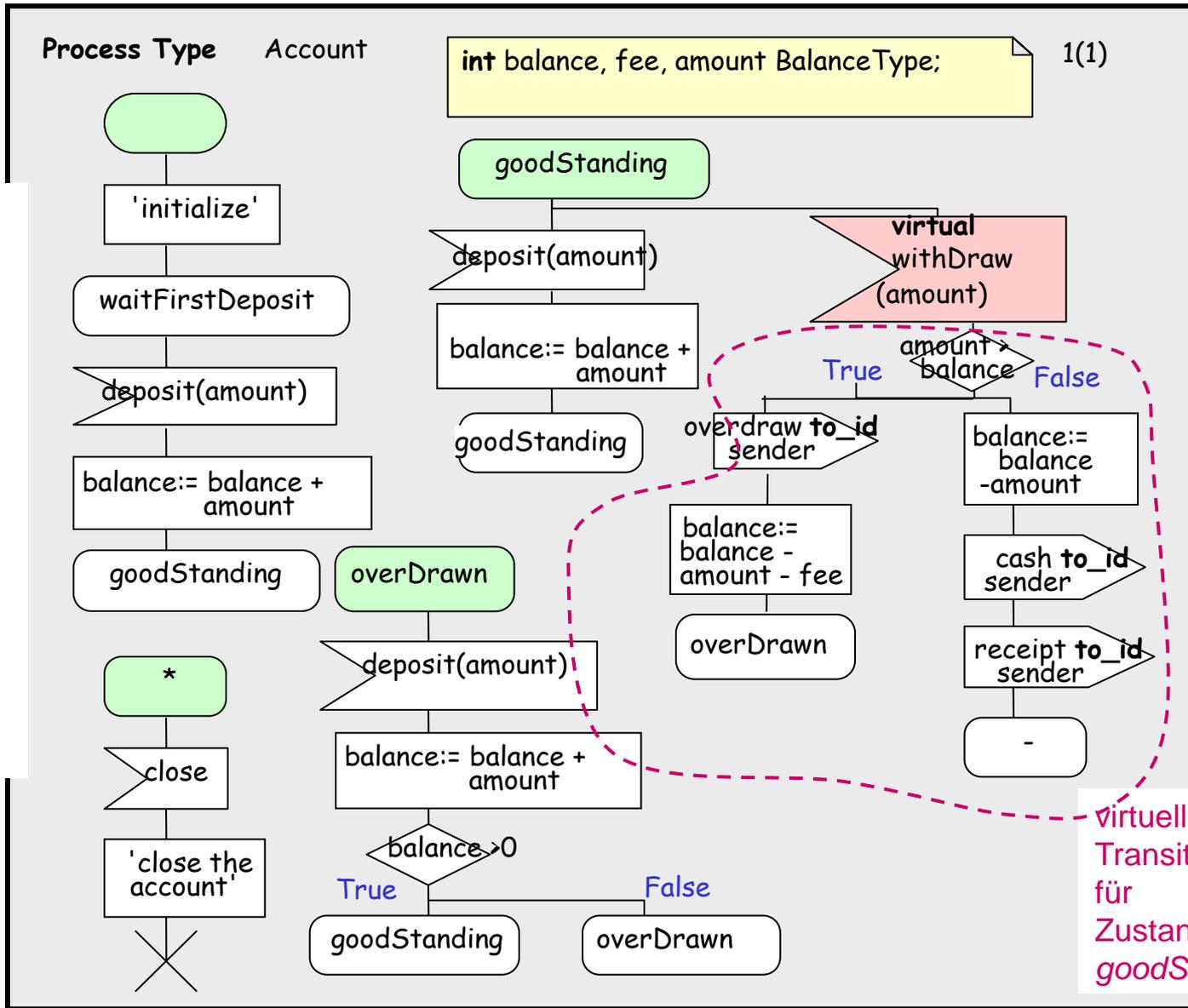
# Redefinition von Transitionen: Zusammenfassung



# Redefinition von Transitionen: Standard-SDL



# Basistyp



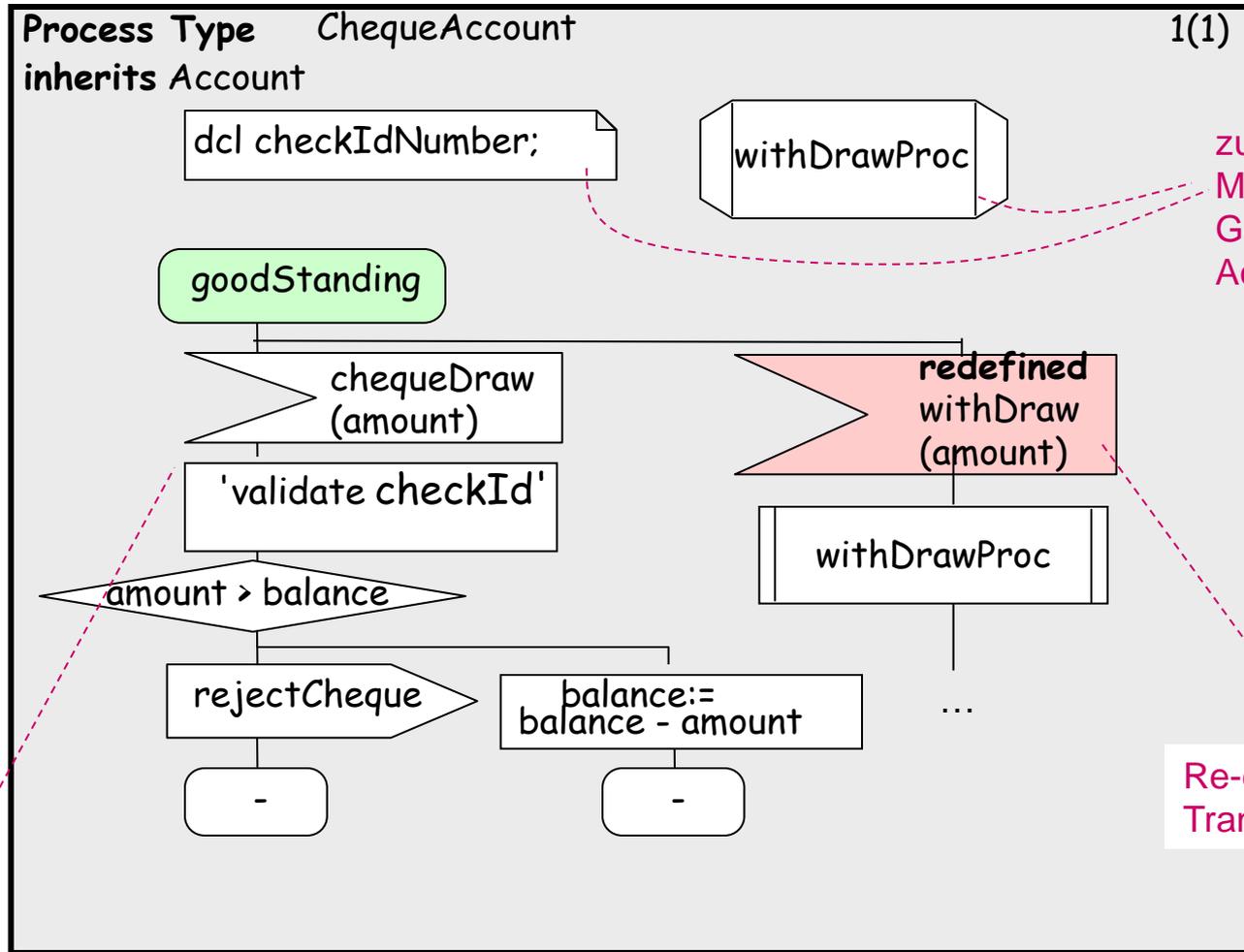
exit

[cash, receipt, overdraw]

[deposit, withDraw, close]

entry

# Ableitung



zusätzliche  
Merkmale  
Gegenüber  
Account

Re-definierte  
Transition

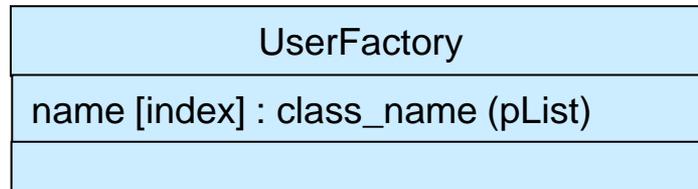
zusätzliche  
Transition

zusätzliche  
Signale für geerbte Gates

## 9. *SDL-Konzepte* (Präzisierung, 2. Teil)

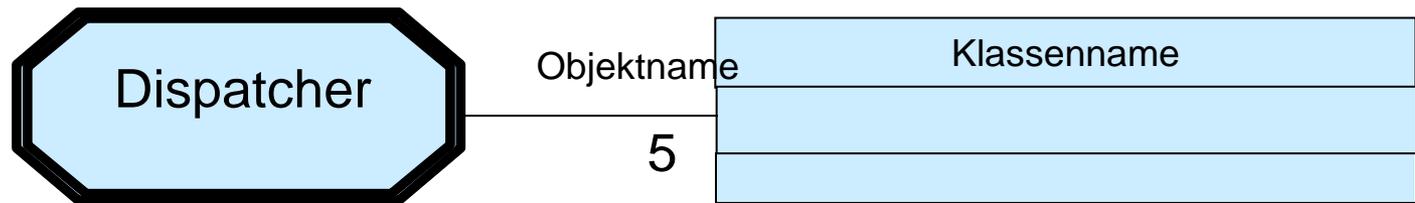
1. Ersetzungskonzept: Shorthand-Notation
2. Ersetzungskonzept: Priorisierter Input
3. Prozeduren, Ersetzungskonzept: Remote-Prozeduren
4. Programm-Ausnahmen (SDL-2000-Konzept)
5. Spezialisierung von Zustandsautomaten
6. Lokale Objekte, Semaphore

- In Klassendeklaration



- In aktiven Klassen

(1) Attributdeklaration



(2) Objekterzeugung (Muss in Starttransition erfolgen)



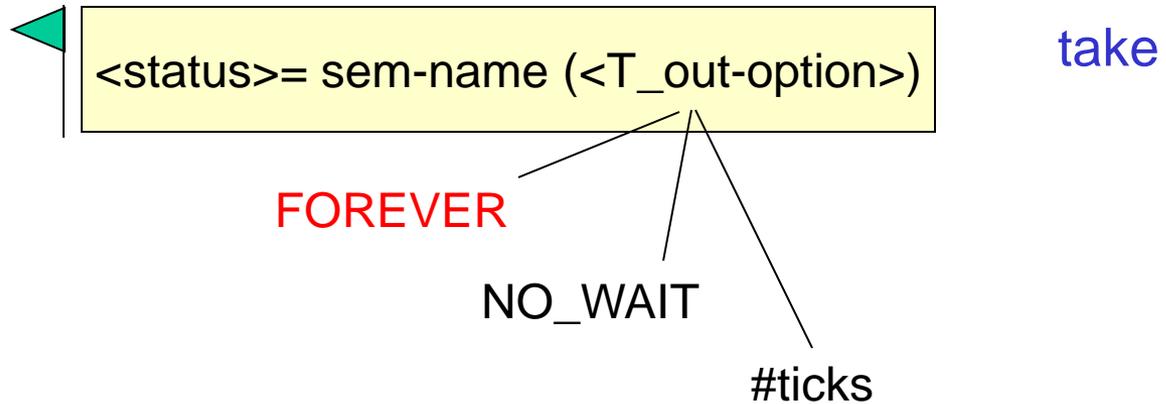
# Semaphore-Deklarationen

## drei Semaphore-Varianten

- **BINARY** b ( [ PRIOR, FIFO ],  
[ INITIAL\_EMPTY, INITIAL\_FULL ] )
- **MUTEX** m ( [ **PRIOR**, FIFO ],  
[ DELETE\_SAVE ],  
[ INVERSION\_SAVE ] )
- **COUNTING** c ( [ PRIOR, FIFO ],  
[ InitialWert(int) ] )

Im Detail leider nicht beschrieben

# Semaphore-Nutzung



- RTDS\_OK: Token wurde erfolgreich entnommen
- RTDS\_ERROR: kein Token vorgefunden bzw. Timeout



# *10. SDL und Next-Event-Simulatoren*

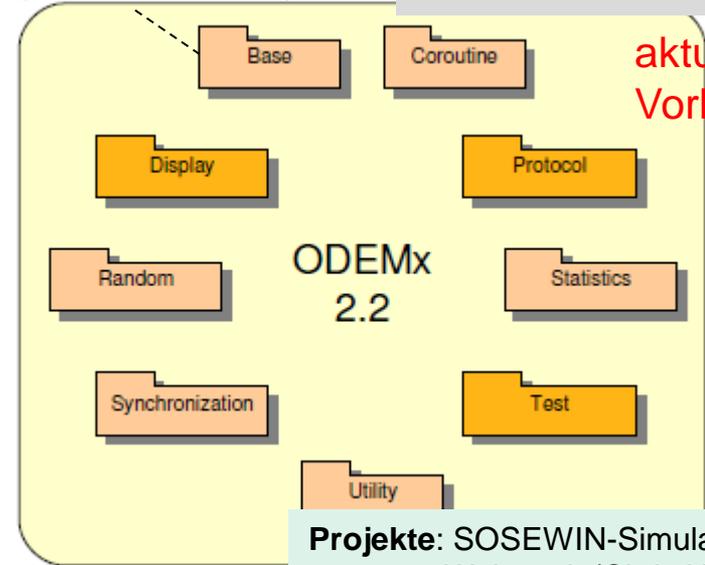
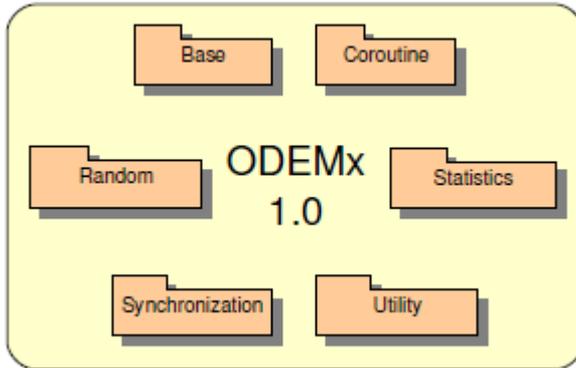
1. ODEMX-Versionen und aktuelle Projektlage
2. Protokollentwicklung für Maschennetzwerke (SOSEWIN)
3. Die ODEMX-Bibliothek in der Version 3.3

# ODEMx-Versionen

Ereignis-Behandlung+

SA Ronald Kluth

DA Ralf Gerstenberger



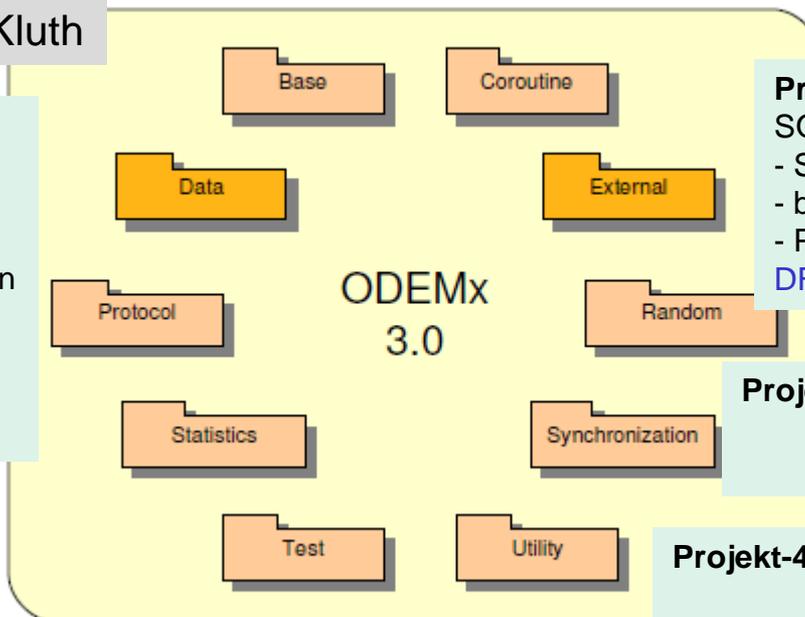
aktuelle Vorlesung

**Projekte:** SOSEWIN-Simulation  
Walzwerk (Chris Helbing)  
Vergüterei (Ingmar Eveslage)

DA Ronald Kluth

## Projekt-2 [Ahrens]

Walzwerk  
Vergüterei  
bei Einbindung in EMS  
Kopplung mit Animatoren  
Einsatz Prod. Planung  
Paradebeispiel für  
Industrial WFL  
BMBF-Förderung???



## Projekt-1: [Eveslage, Brumbulli]

SOSEWIN-Simulation  
- SDL/RT+ASN.1 → ODEMx, ns-3, Zielcode  
- bei Einbindung in EMS  
- Paradebeispiel für Scientific WFL  
DFG-SPP, BMBF-SOSEWINextended

## Projekt-3 [Blunk]: Workflow-Sprache

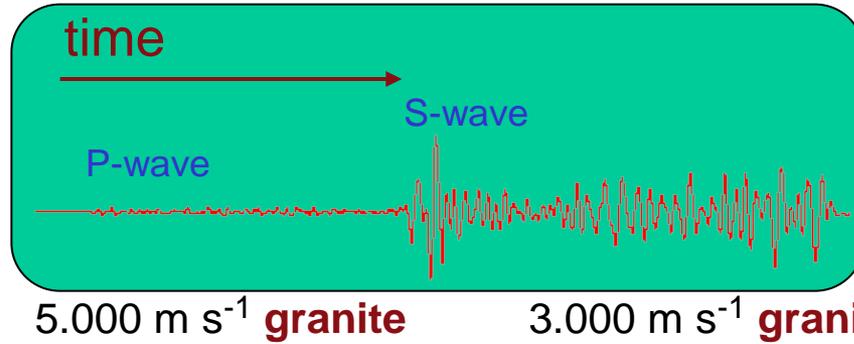
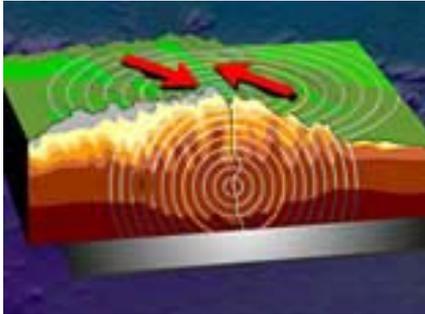
Meta-modelbasiert  
Laufzeitsystem in ODEMx

**Projekt-4 [???]:** Parallelisierung von Simulationexp.  
Hochwasser

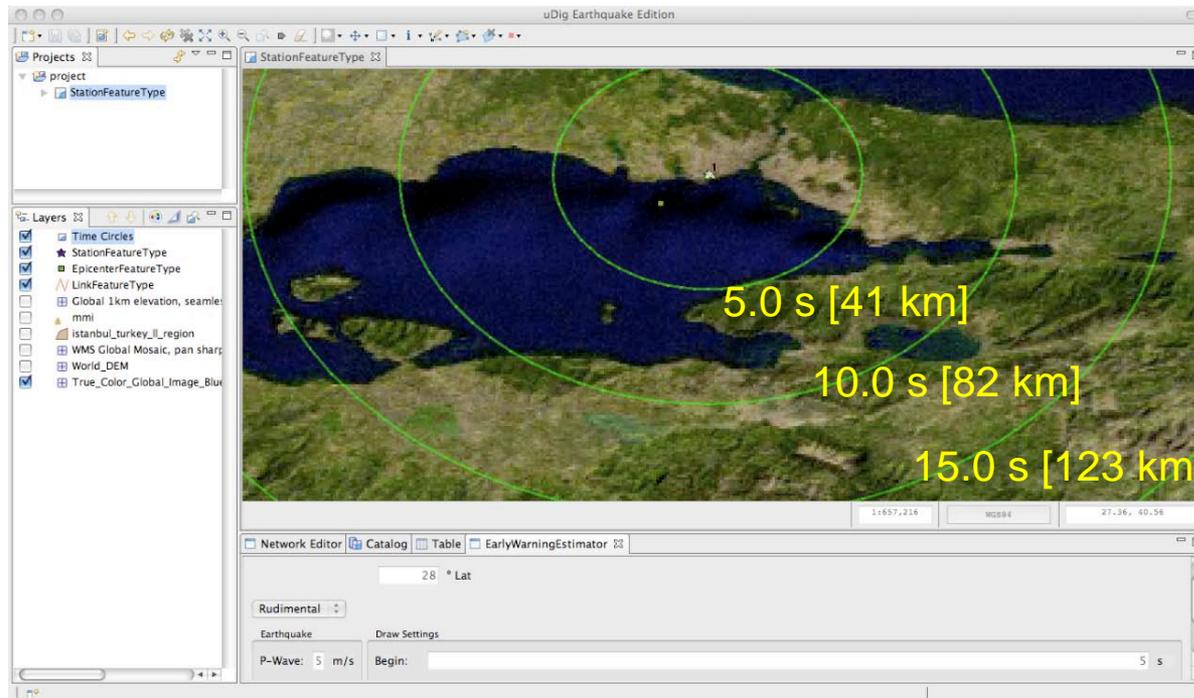
# 10. *SDL und Next-Event-Simulatoren*

1. ODEMX-Versionen und aktuelle Projektlage
2. Protokollentwicklung für Maschennetzwerke (SOSEWIN)
3. Die ODEMX-Bibliothek in der Version 3.3

# Physical Base of Early Warning



acceleration [cm s<sup>-2</sup>]  
of one  
of 3 dimensions



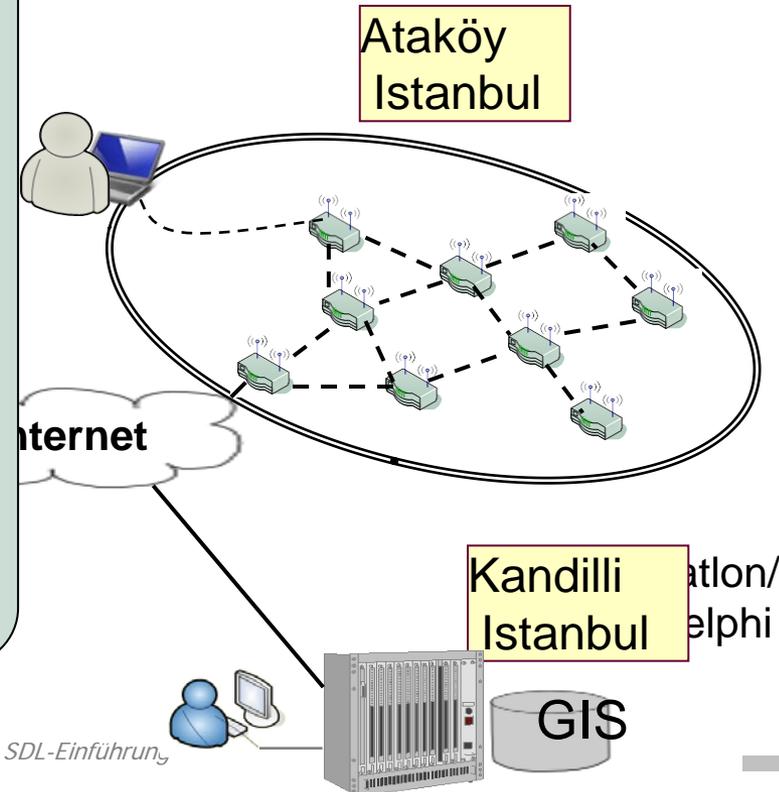
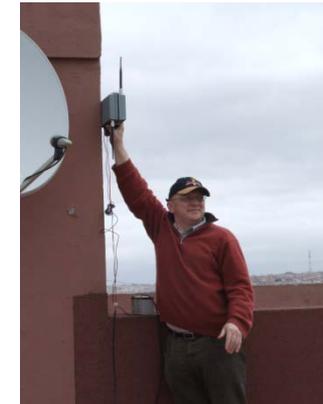
# SOSEWIN – EEWS Prototype in Istanbul

## network characteristics

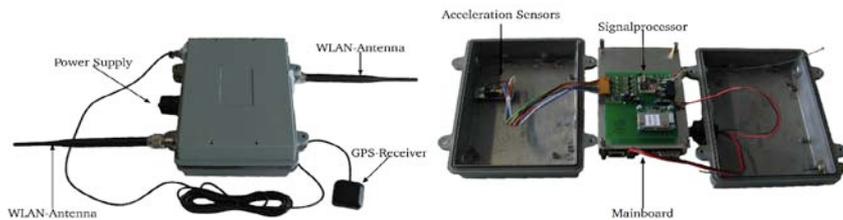
- 20 nodes
- wireless
- meshed, multi-hop
- distance: 100m – 200m
- area: 600m x 800m
- decentralized management
- low-cost seismometer: 300 €/ node

## node characteristics

- mini PC
- AD converter
- GPS unit
- Linux OS / OpenWRT
- communication protocol stack
- middleware (general services)
- application (signal analysis, alarm)

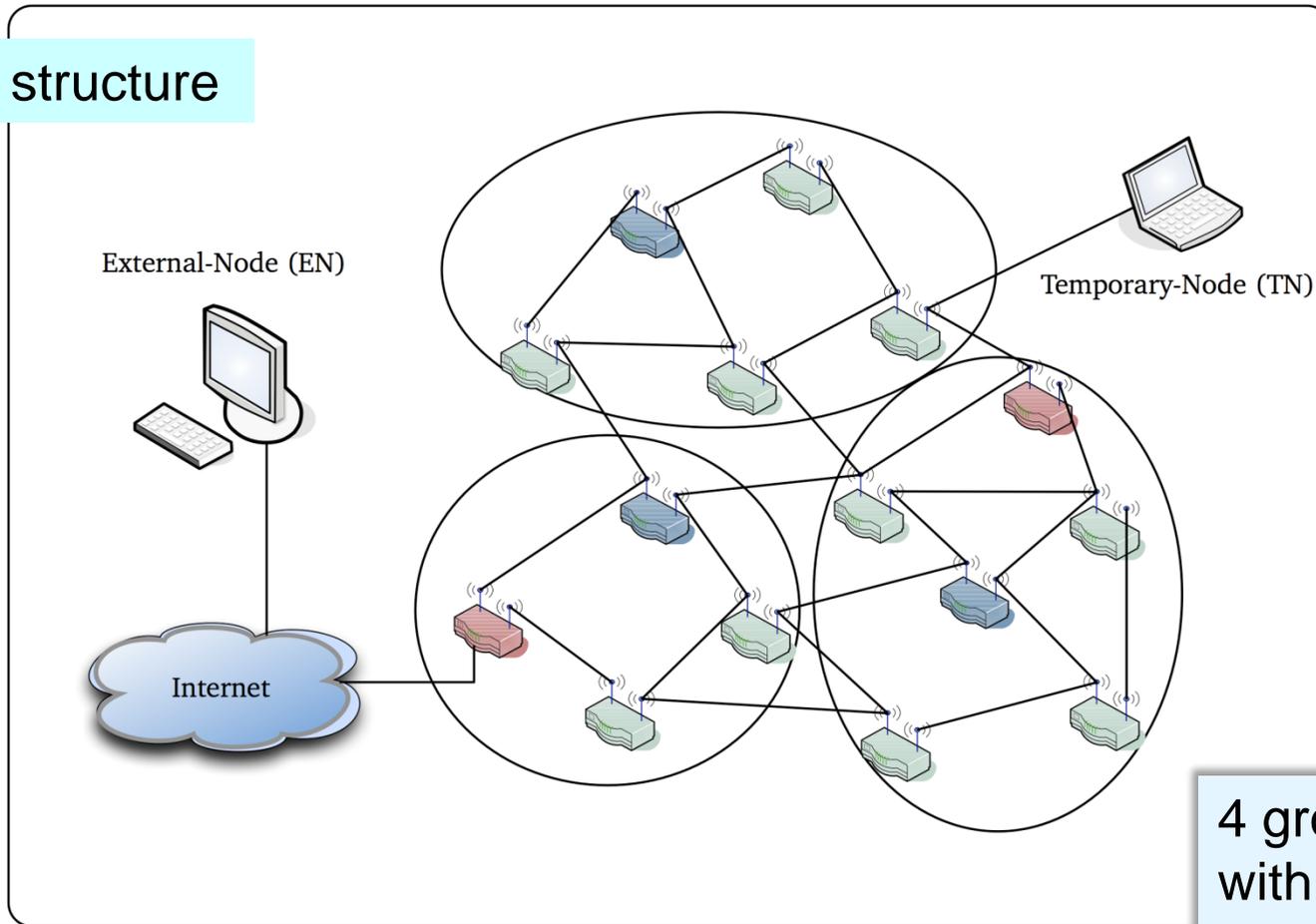


## Nach 2 Jahren Outdoor-Einsatz, Juni 2010

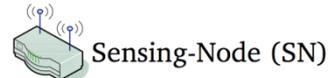
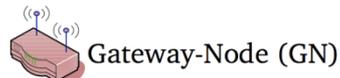
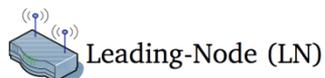


# SOSEWIN – Net Topology

group structure

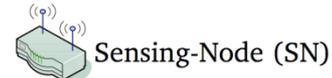
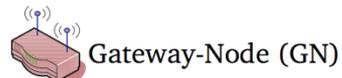
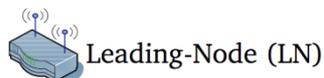
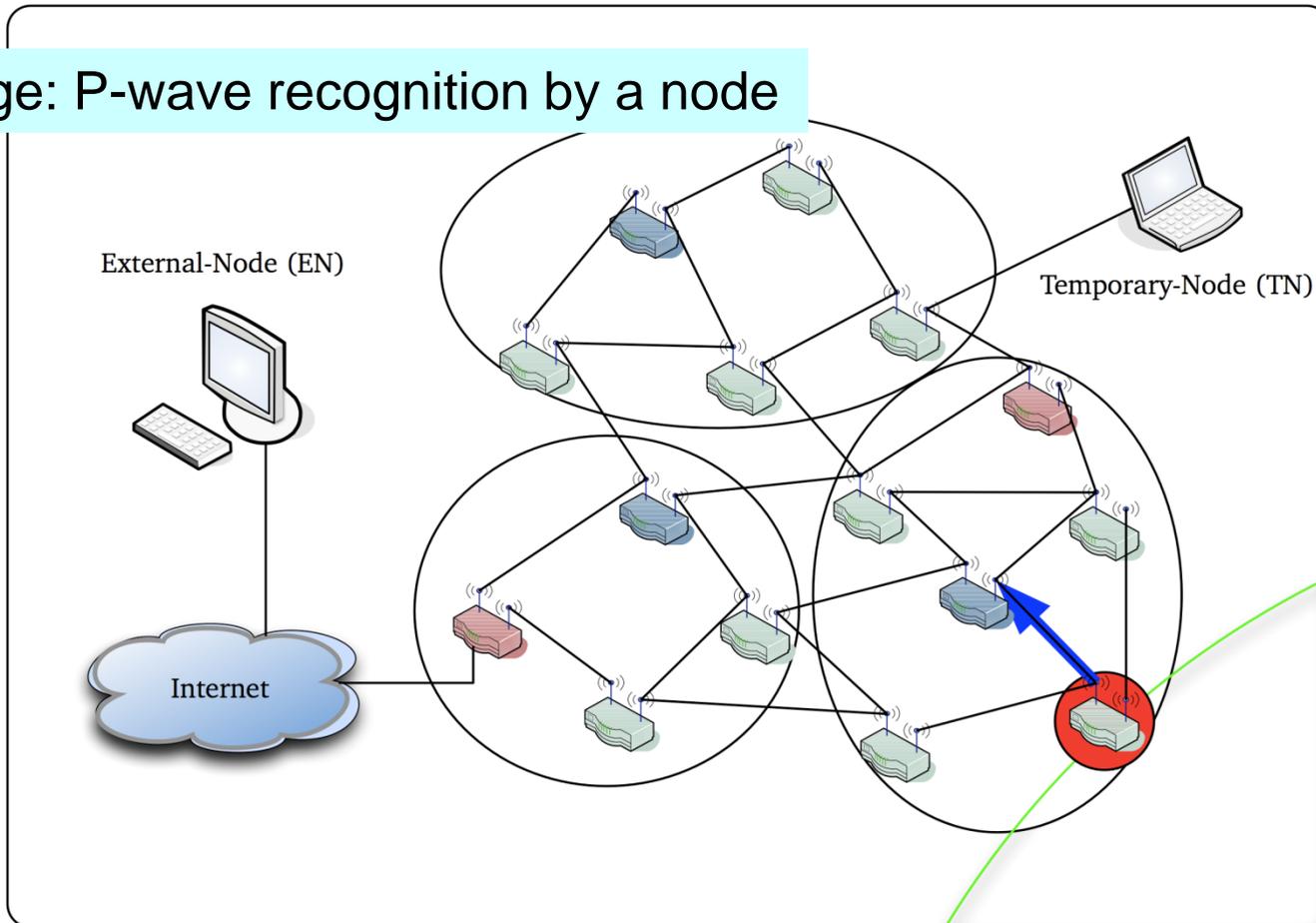


4 groups  
with 5 nodes



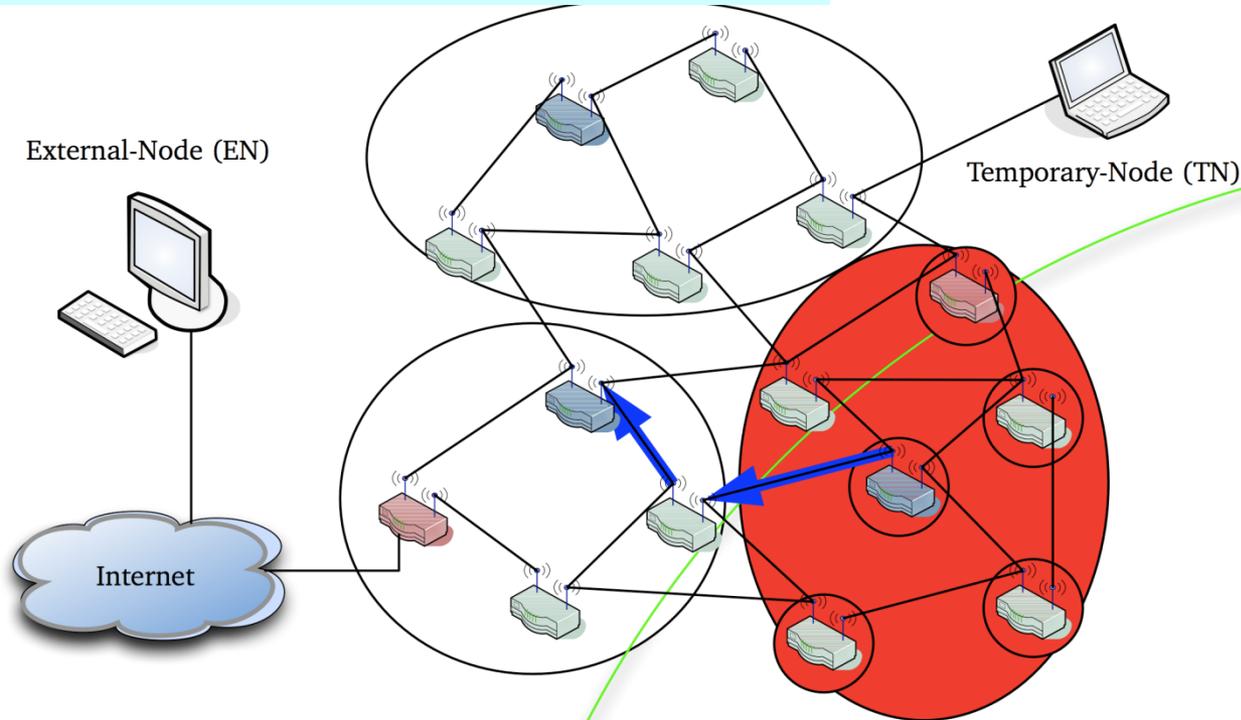
# SOSEWIN – 3-Stage Alarming Protocol

## 1st Stage: P-wave recognition by a node



# SOSEWIN – 3-Stage Alarming Protocol

2nd Stage: 50% of the members by a group



3rd Stage: 50% of all groups → system alarm



Leading-Node (LN)

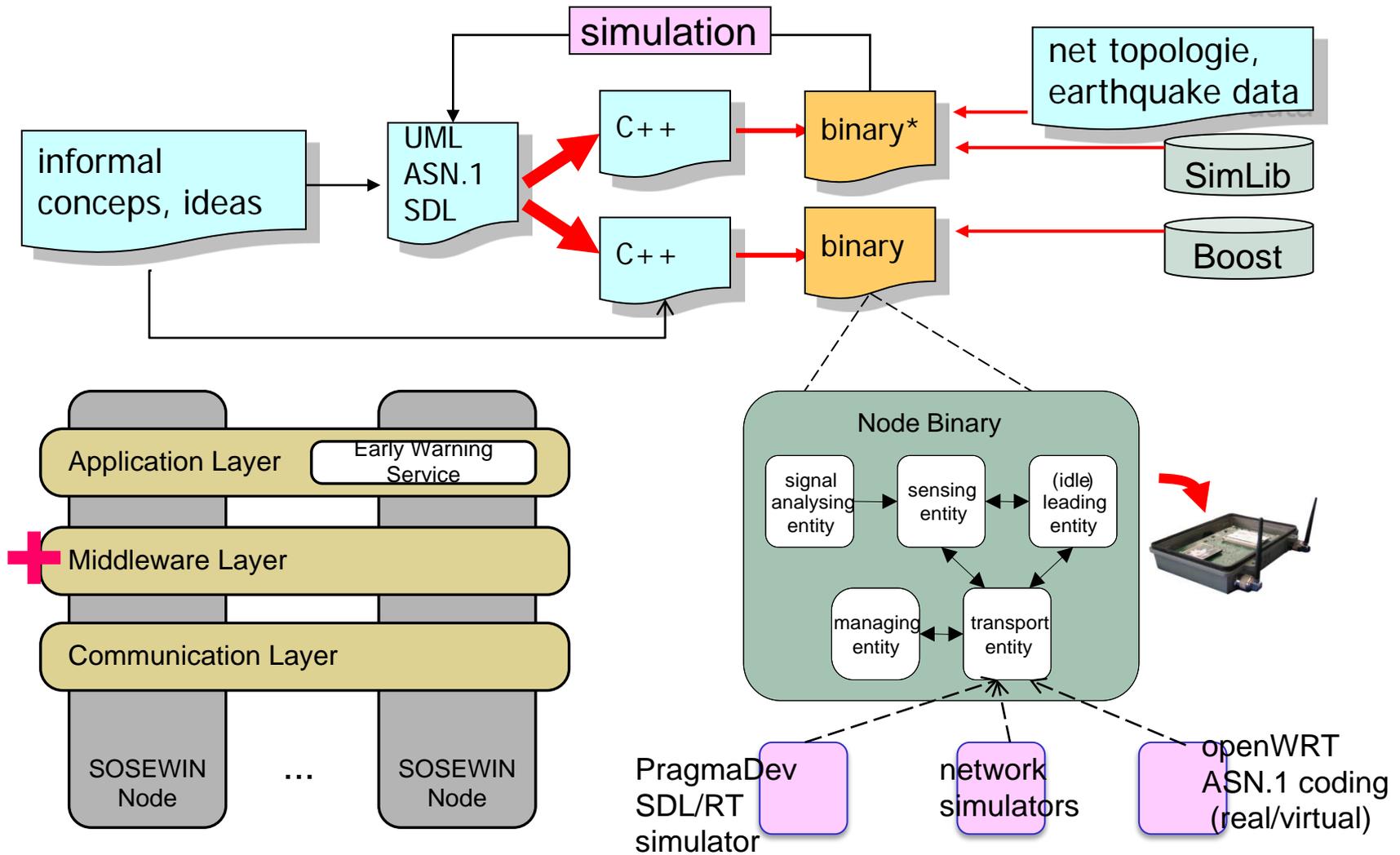


Gateway-Node (GN)



Sensing-Node (SN)

# Our Model-Driven Approach

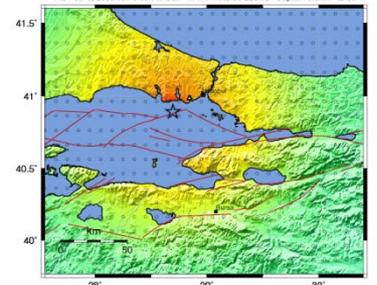


in dependence of the communication infrastructure

# Experiment Management System

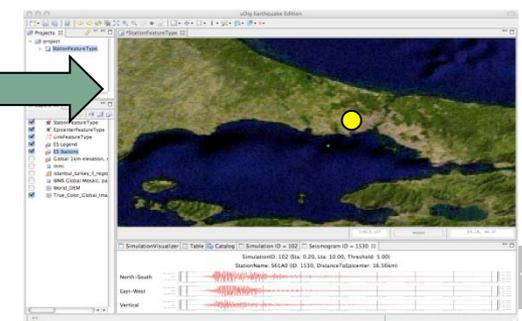
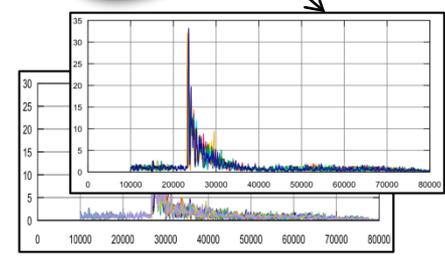
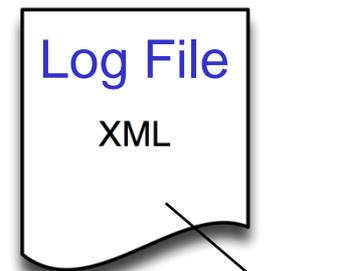
## Executables

GFZ ShakeMap : Location description default. Created by Network Editor.  
Thu Feb 5, 2009 05:16:06 PM SST M 7.4 N40.89 E28.70 Depth: 10.0km ID:60

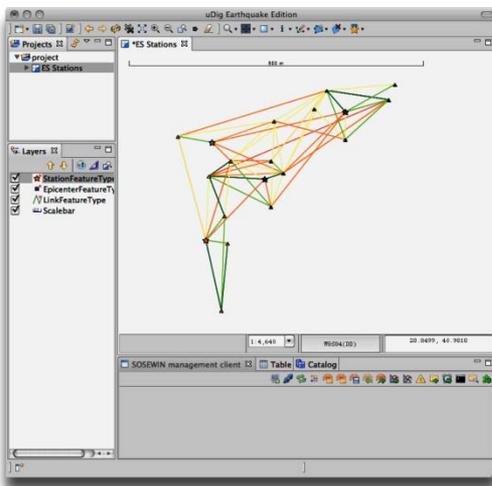
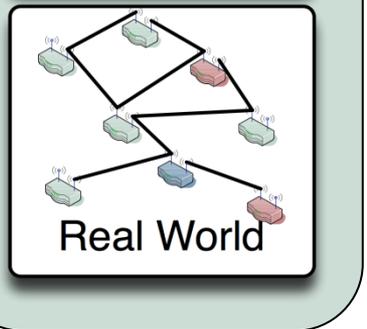
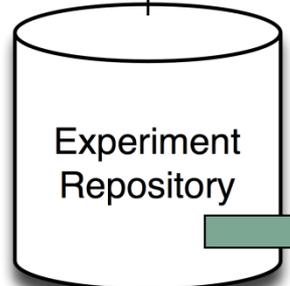
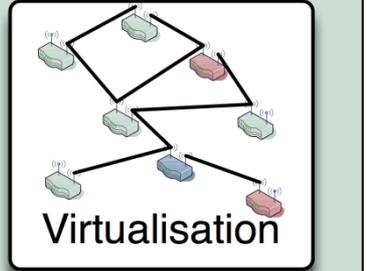
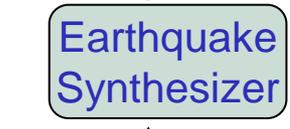
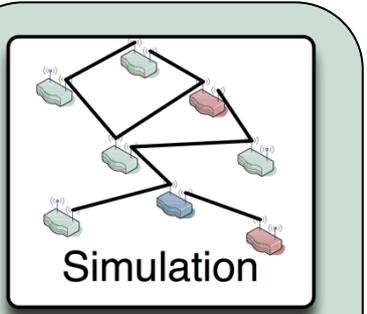
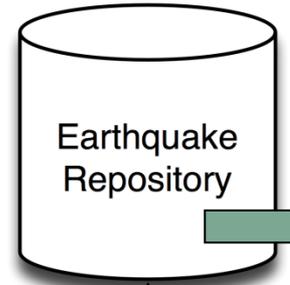


Map Version 1 Processed Thu Feb 5, 2009 06:10:13 PM SST - NOT REVIEWED BY HUMAN

Intensity	Not felt	Weak	Light	Moderate	Strong	Very strong	Severe	Violent	Extreme
MMI Scale	none	none	Very light	Light	Moderate	Moderate/Heavy	Heavy	Very Heavy	>10
Peak Acceleration	<0.7	0.7-1.0	1.0-2.0	2.0-5.0	5.0-10	10-20	20-50	50-100	>100
Peak Velocity	<0.3	0.3-1.1	1.1-3.4	3.4-11	11-33	33-100	100-300	300-1000	>1000
Maximum Displacement	I	II	III	IV	V	VI	VII	VIII	IX

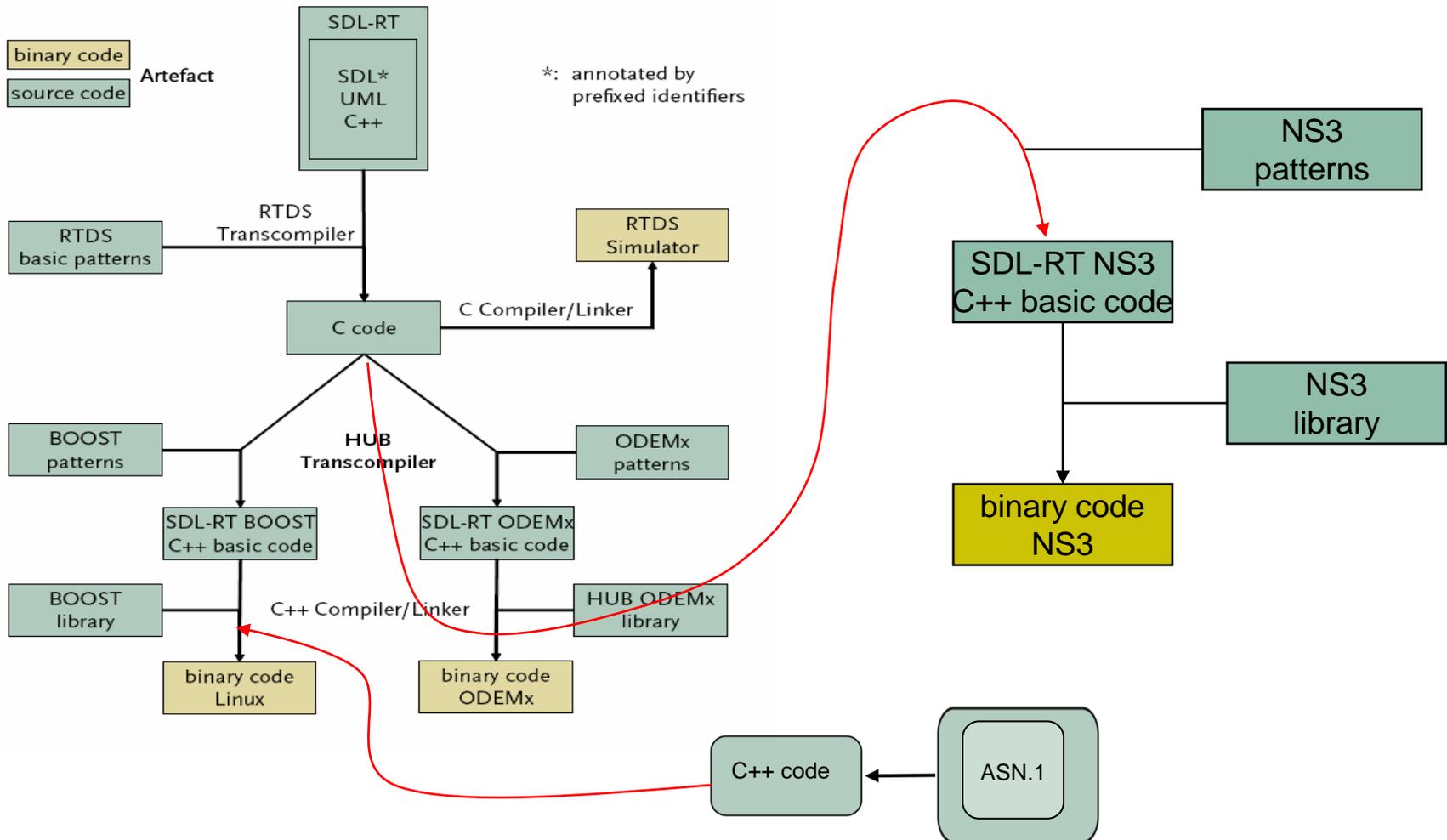


Simulation ID	Name	Simulation Time	Queue	Status	Start Time	End Time	Duration
Completed 210	msw_earthquake_simulation_01	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 211	msw_earthquake_simulation_02	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 212	msw_earthquake_simulation_03	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 213	msw_earthquake_simulation_04	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 214	msw_earthquake_simulation_05	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 215	msw_earthquake_simulation_06	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 216	msw_earthquake_simulation_07	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 217	msw_earthquake_simulation_08	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 218	msw_earthquake_simulation_09	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 219	msw_earthquake_simulation_10	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 220	msw_earthquake_simulation_11	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 221	msw_earthquake_simulation_12	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 222	msw_earthquake_simulation_13	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 223	msw_earthquake_simulation_14	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00
Completed 224	msw_earthquake_simulation_15	2009-02-12 12:59:03.18	IC	Completed	2009-02-20 12:24:29	2009-02-20 12:27:29	00:03:00

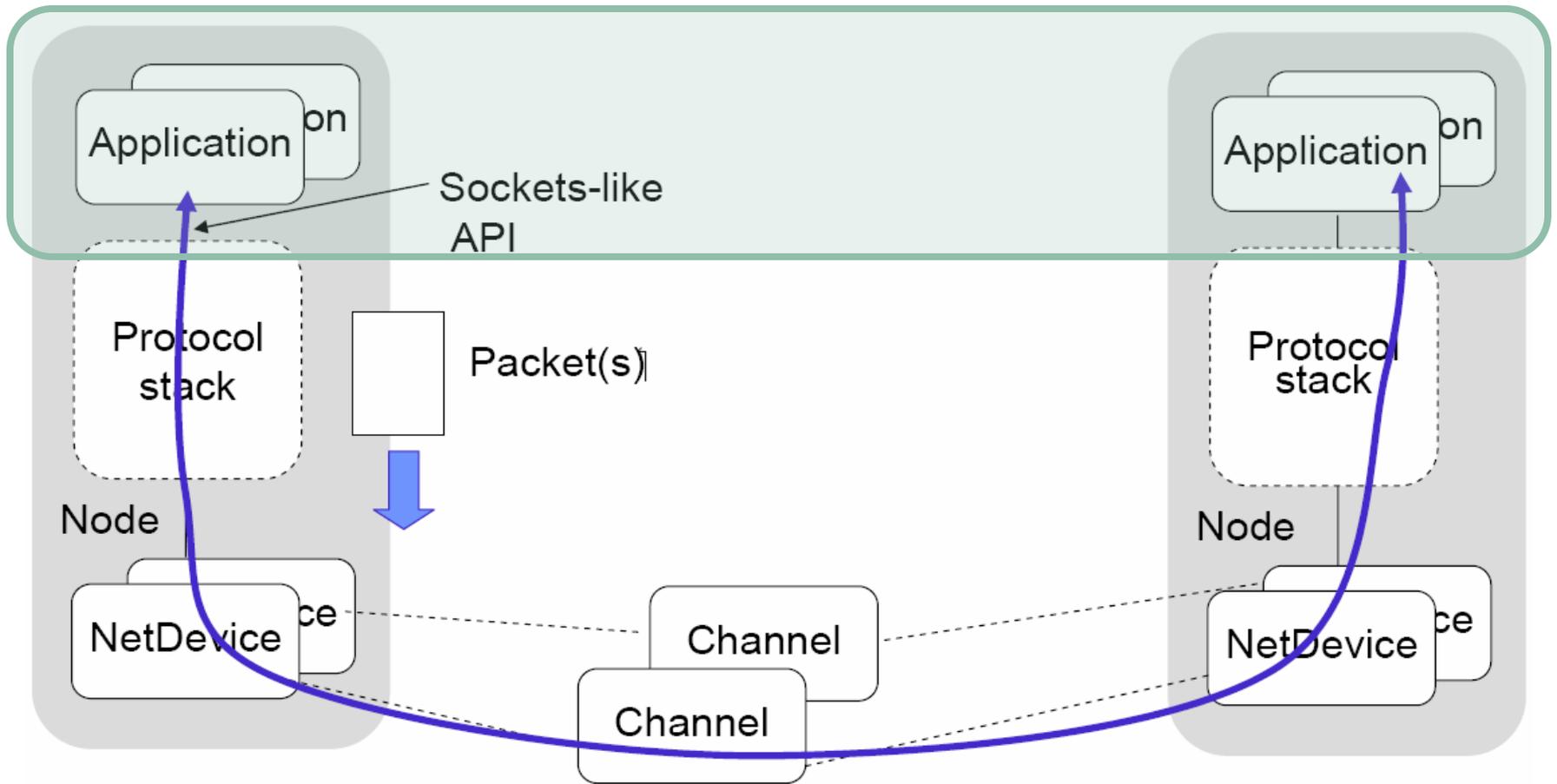


GIS-based editor  
J. Fischer

# Extension of PragmaDevs SDL/RT Compiler



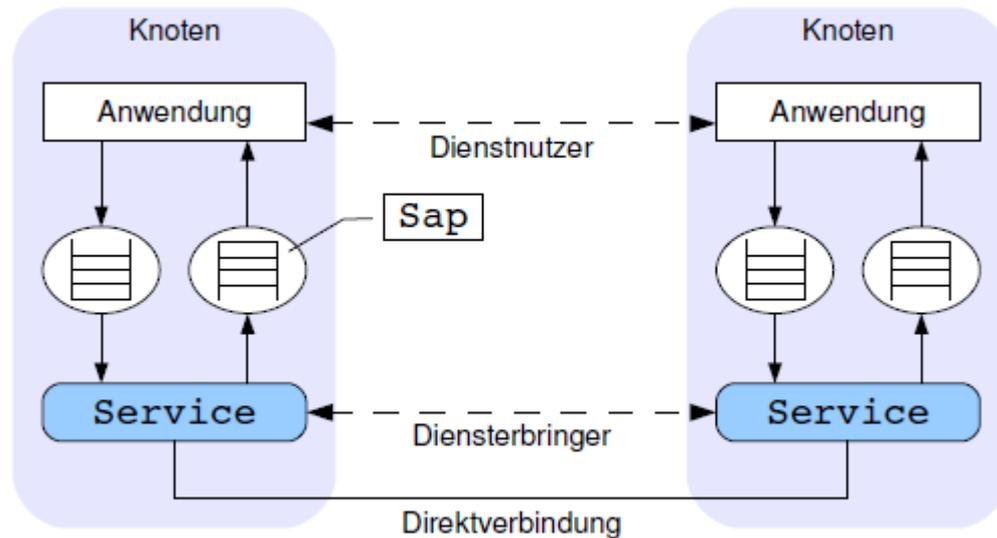
# ns-3 Model Overview



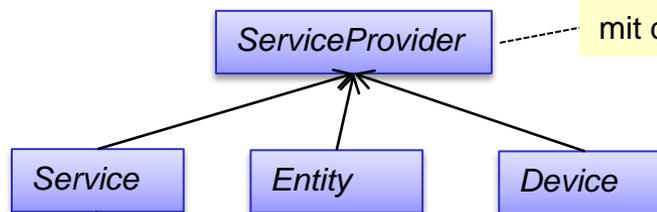
# 10. *SDL und Next-Event-Simulatoren*

1. ODEMX-Versionen und aktuelle Projektlage
2. Protokollentwicklung für Maschennetzwerke (SOSEWIN)
3. Die ODEMX-Bibliothek in der Version 3.3

# Dienstnutzer - Dienstleister



Suche nach Konzepten zur Vorab-Definition von Struktur- und Verhaltenseinheiten



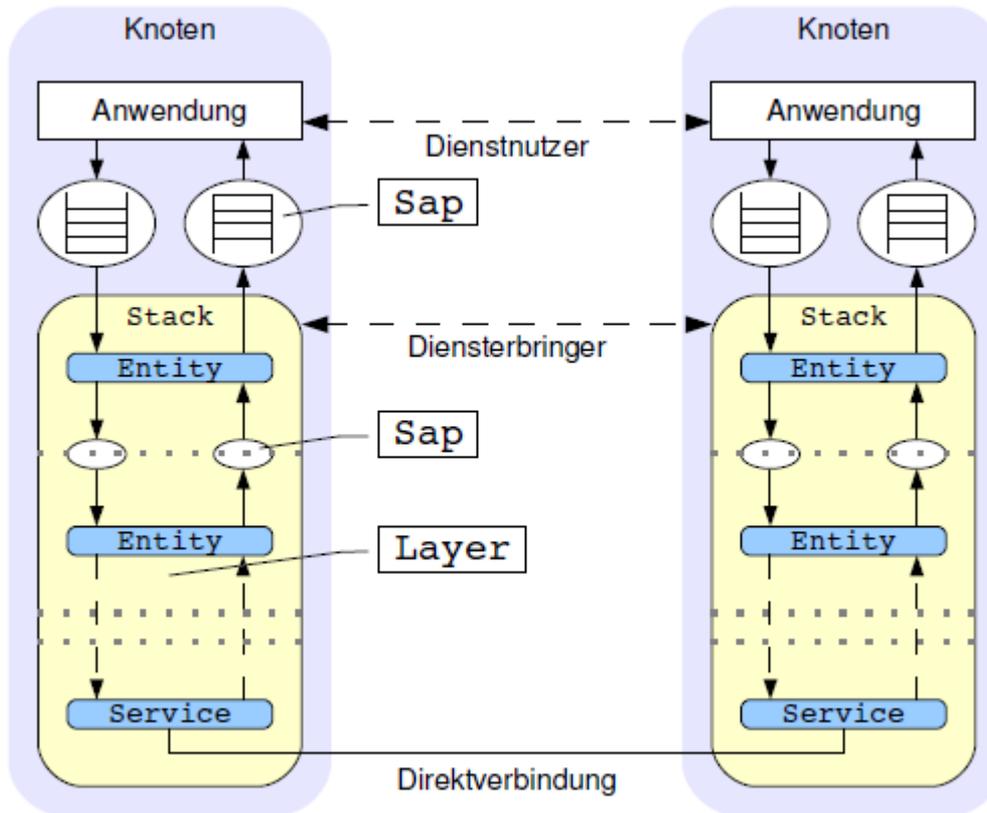
mit dynamischer SAP-Zuordnung (inkl. Puffer)

Kollektion von Entities zur Erbringung eines Dienstes mit Schnittstelldefinition als Abstraktion der darunter liegenden Schichtenfunktionalität

Prototyp für Protocol-Entity

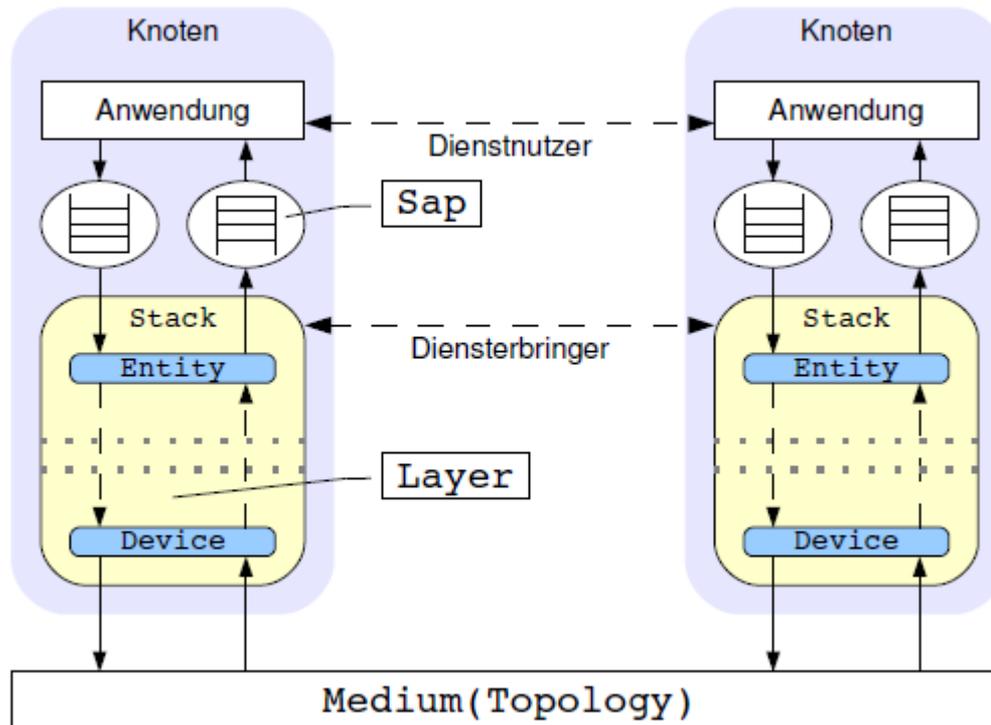
als Interface zu gewählten Übertragungsmedien

# Knoten und Protokoll-Stack



einem Knoten können verschiedene Protokollstacks zugeordnet werden

# Die Klassen im Modul Protocol



Device-Objekte  
Registrieren sich  
beim Medium  
- Zustandkommen von Links  
- bei separater  
Qualitätsbeschreibung  
der Links

Möglichkeit zur Darstellung  
verschiedener Medien für die  
Übertragung  
(WLAN, IP, Satellit)

Möglichkeit zur Darstellung  
von zwei Ereignissen einer  
Übertragung: Beginn, Ende  
(wichtig  
für Nachbildung von Kollisionen,  
Die von Device-Objekten  
erkannt werden)

PDU mit zwei Funktionen:  
-getSize() liefert Länge einer Nachricht  
-clone()

Wir erwarten Sie !

Für alle Prüfungen  
viel Erfolg

