

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

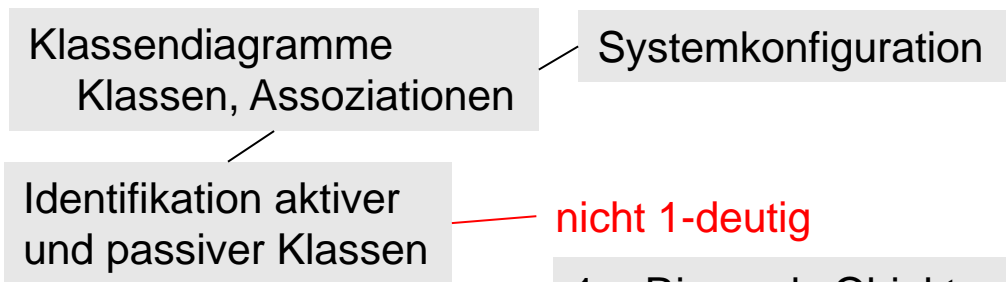
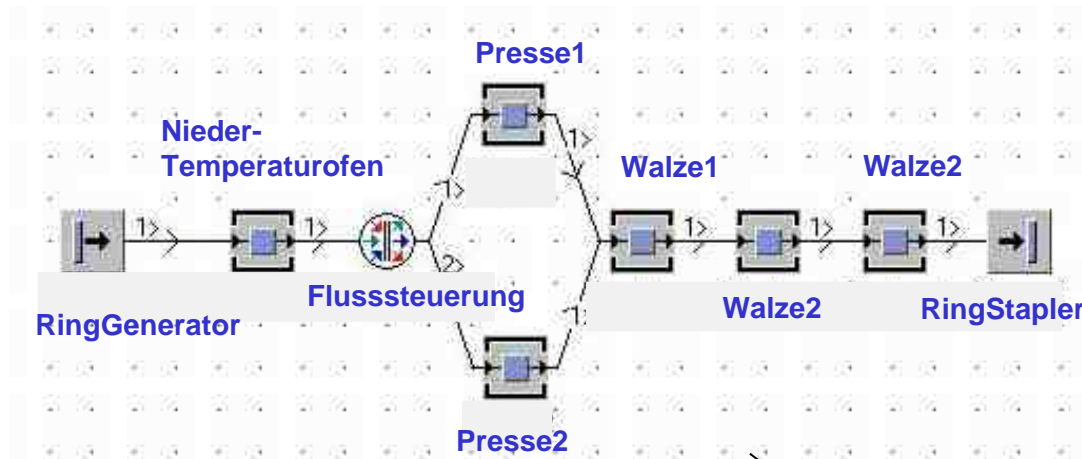
Objektorientierte Simulation ***mit ODEMx***

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

fischer|ahrens|eveslage|blunk@informatik.hu-berlin.de

4. Ereignisse in ODEMx

Grundsatzentscheidungen bei der Modellierung

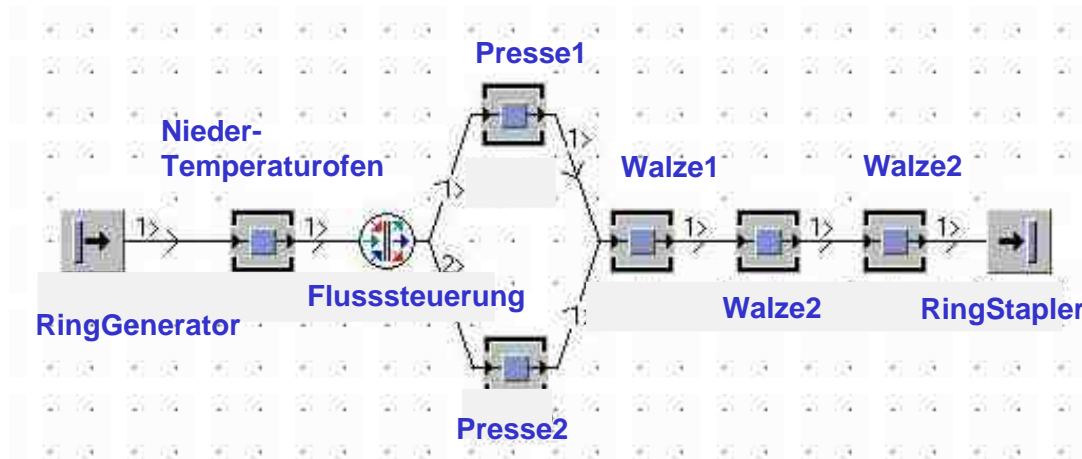


Entscheidung:
häufig bestimmt durch die Konzepte
der verwendeten Modellierungssprache

ODEMx als universelle Simulationsbibliothek
erlaubt alle Varianten

- 1. Ringe als Objekte aktiver Klassen ~GPSS
Stationen als Objekte passiver Klassen
- 2. Ringe als Objekte passiver Klassen ~SDL
Stationen als Objekte aktiver Klassen
- 3. Ringe u. Stationen als Objekte
aktiver Klassen

Grundsatzentscheidungen bei der Modellierung (Forts.)



Verhaltensmodellierung:
- ereignisorientiert
- processorientiert

Entscheidung:

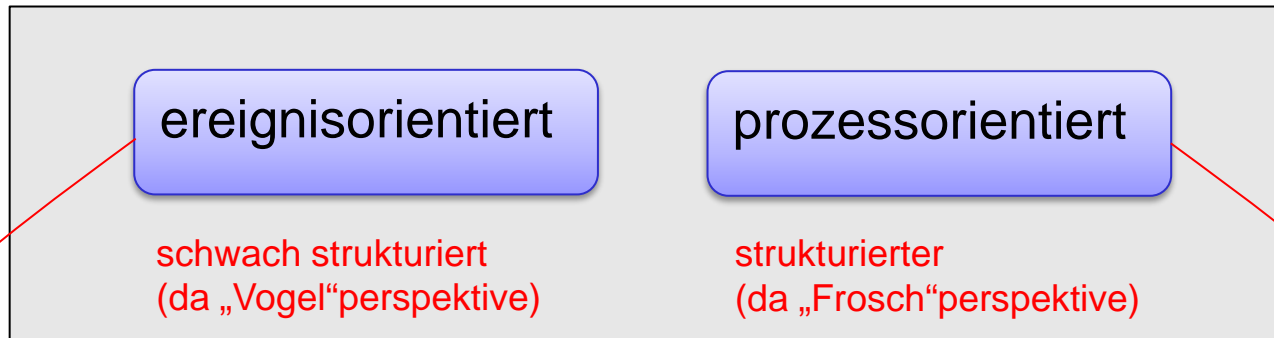
häufig bestimmt durch die Konzepte der verwendeten Modellierungssprache

ODEMx als universelle Simulationsbibliothek erlaubt alle Varianten

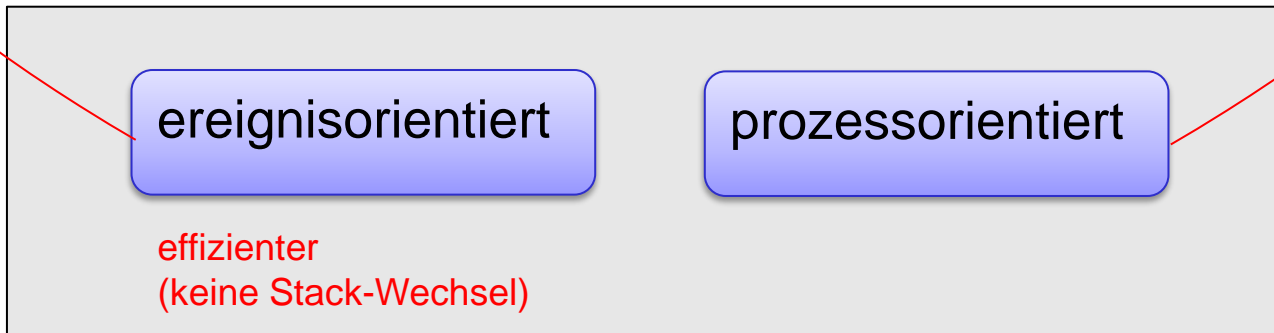
→ Frage nach Vor- und Nachteilen

Verhaltensmodellierung

Modellbeschreibung (ODEMx)



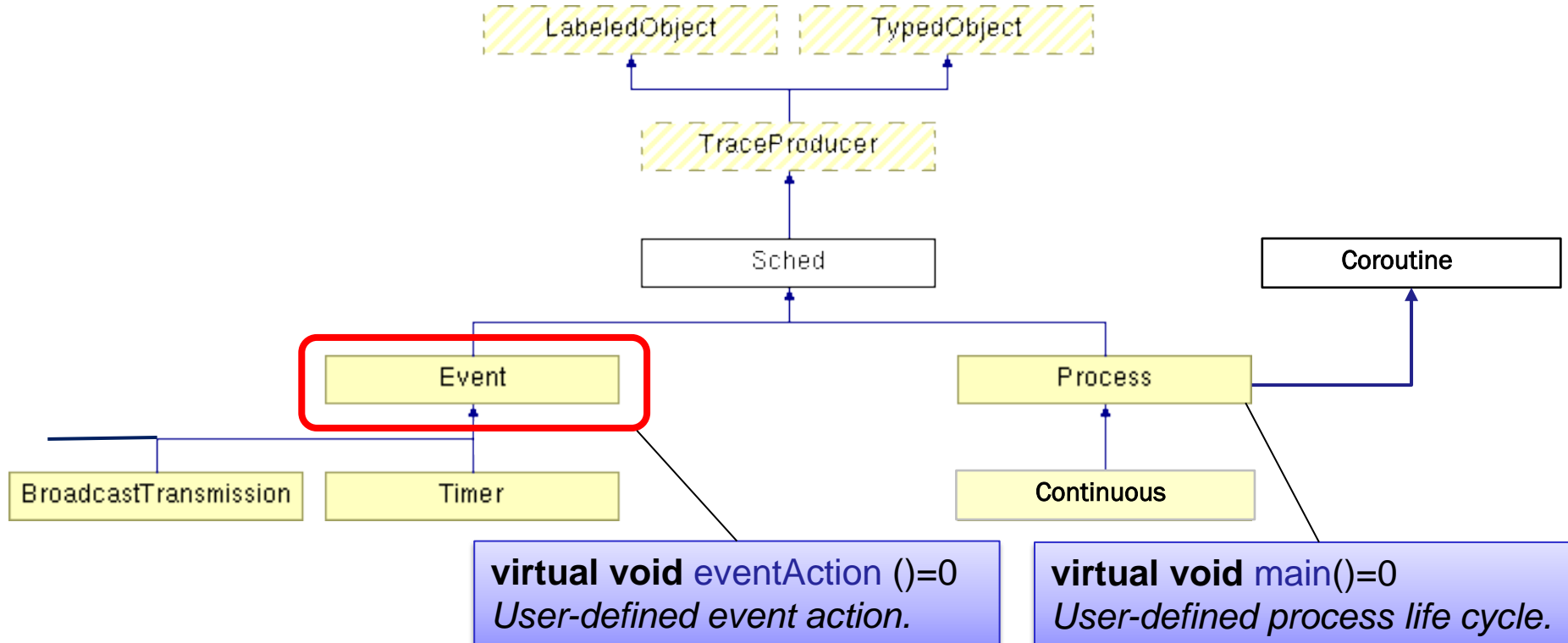
Modellausführung (ODEMx)



Sched-Spezialisierungen (zur Erinnerung)

Ausschnitt der ODEMX-Klassenhierarchie

Basisklassen: Sched, Coroutine



Ereignisfolgenbildung (Event-Scheduling)

void `schedule` ()
Trigger the event at current simulation time (LIFO)

void `scheduleIn` (SimTime t)
Trigger the event in (relative) time t (LIFO)

void `scheduleAt` (SimTime t)
Trigger the event at (absolute) time t (LIFO)

void `scheduleAppend` ()
Trigger the event at current simulation time (FIFO)

void `scheduleAppendIn` (SimTime t)
Trigger the event in (relative) time t (FIFO)

void `scheduleAppendAt` (SimTime t)
Trigger the event at (absolute) time t (FIFO)

void `removeFromSchedule` ()
Remove the event from the execution list

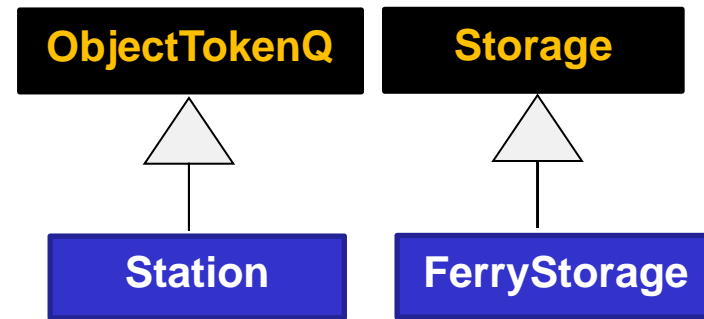
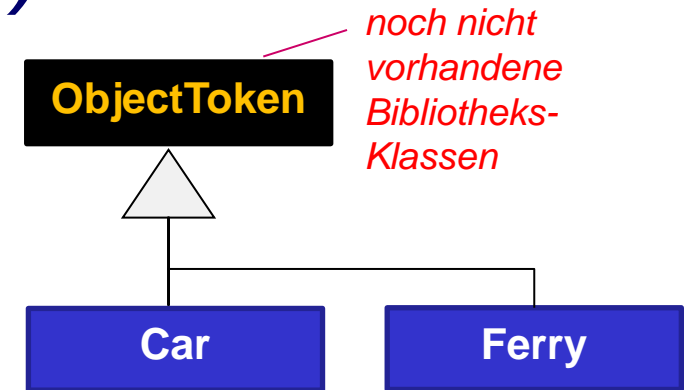
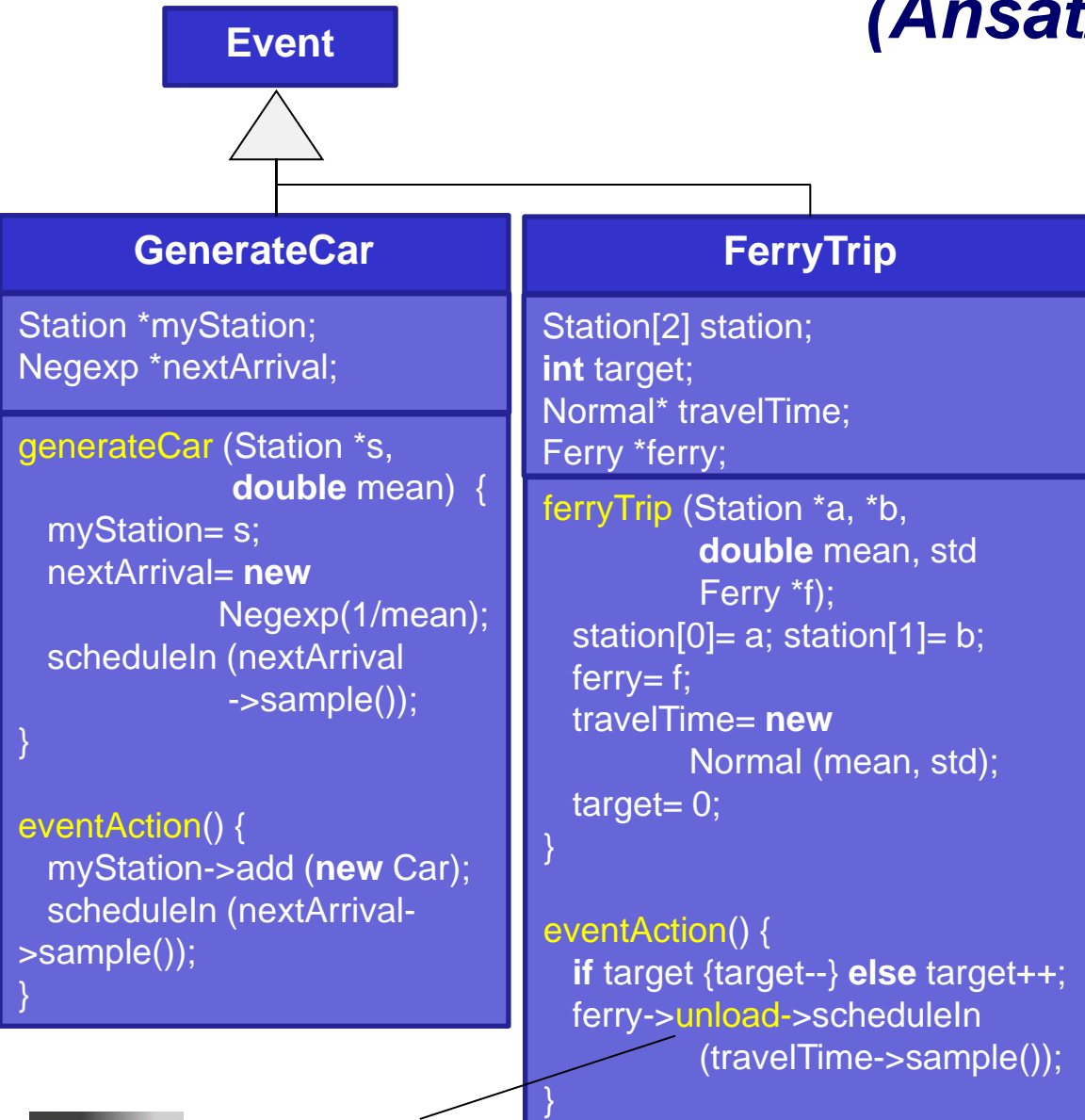
virtual Priority `getPriority` () **const**

Get priority.

virtual Priority `setPriority` (Priority newPriority)

Set new priority

Ereignisorientierte Modellierung: Autofähre (Ansatz)



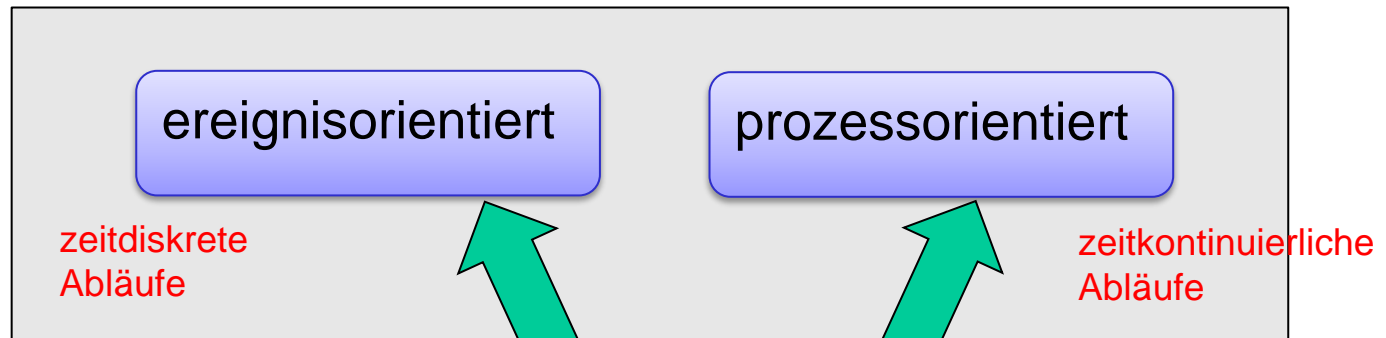
- weitere Ereignisse:
- UnloadCar,
 - LoadCar,
 - CloseWait
 - CloseTrip

UnloadCar-Objekt

Objektorientierte Simulation mit ODEMX

Idealerer Ansatz

- Modellausführung (ODEMx)

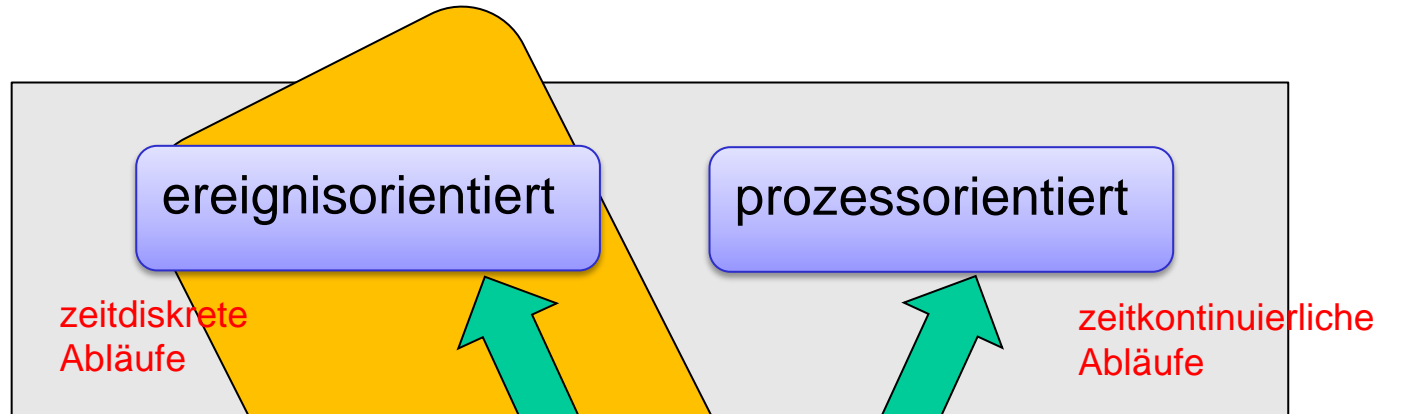


- Modellbeschreibung
(mit geeigneter Modellierungssprache)

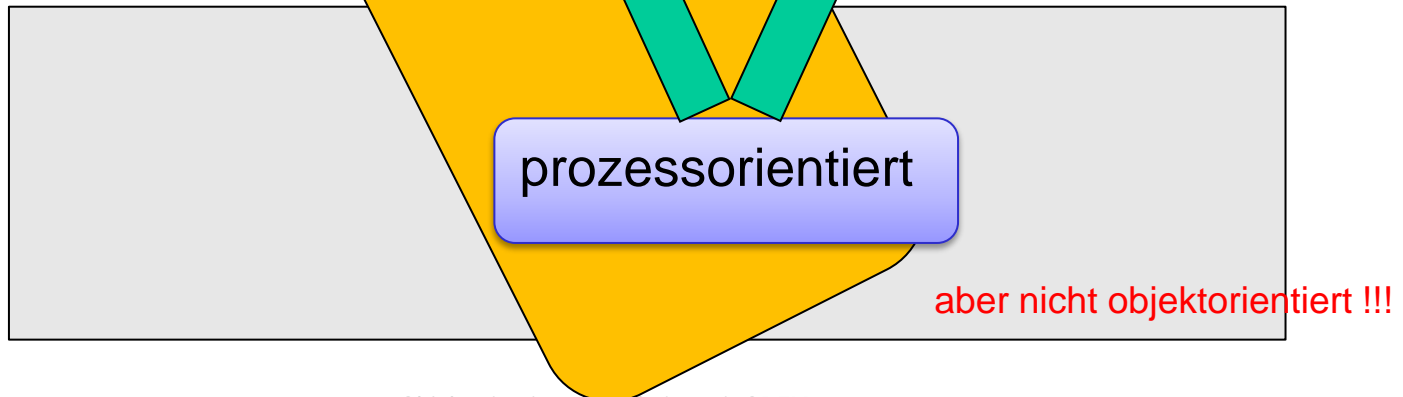


Historischer Ansatz

- Modellausführung (Assembler-Bibliothek)



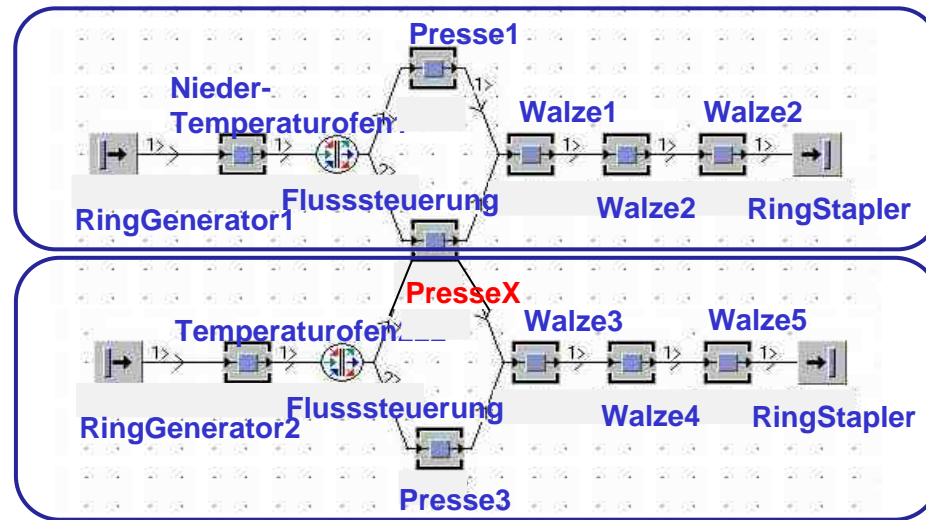
- Modellbeschreibung
(mit geeigneter Modellierungssprache: GPSS)



5. GPSS

1. Grundphilosophie
2. Block/Stations-Übersicht

Aktive und stationäre Modelleinheiten (Entitäten)



Transaction (Transaktion, Aktivator, Objekt-Token) mit

- ausbaufähiger Attributstruktur und
- Verhaltensbeschreibung

als Angabe der anzulaufenden Stationen als Lebenslauf, ausgehend von einer Generator-Station

sehr
eingeschränkt
referenzierbar

Block (Station, Lokalisierung von Aktivitäten) mit

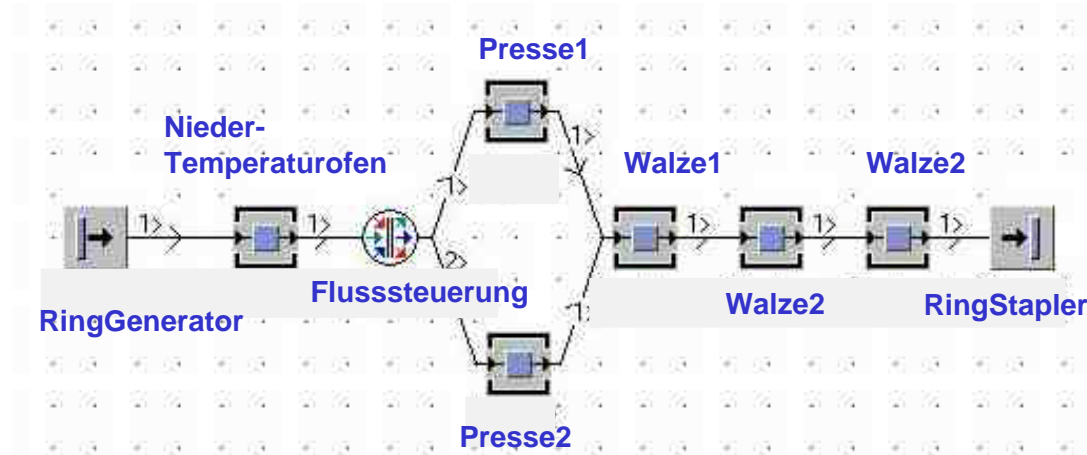
- fester, aber z.T. mit ausdrucksstarken Parametrisierungen
- zeitliche Verzögerung, Attributänderung, Synchronisation

global
1-deutig
referenzierbar

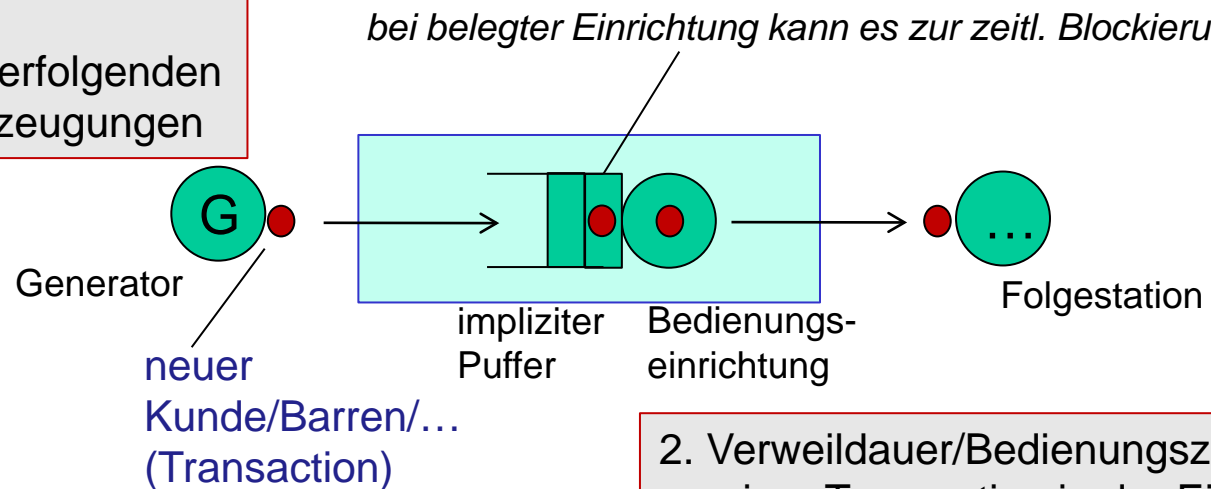
Deklarations-, Initialisierungs- u. Steueranweisungen

- sehr viele Modellverwaltungsstrukturen sind vordefiniert

Verbrauch von Modellzeit

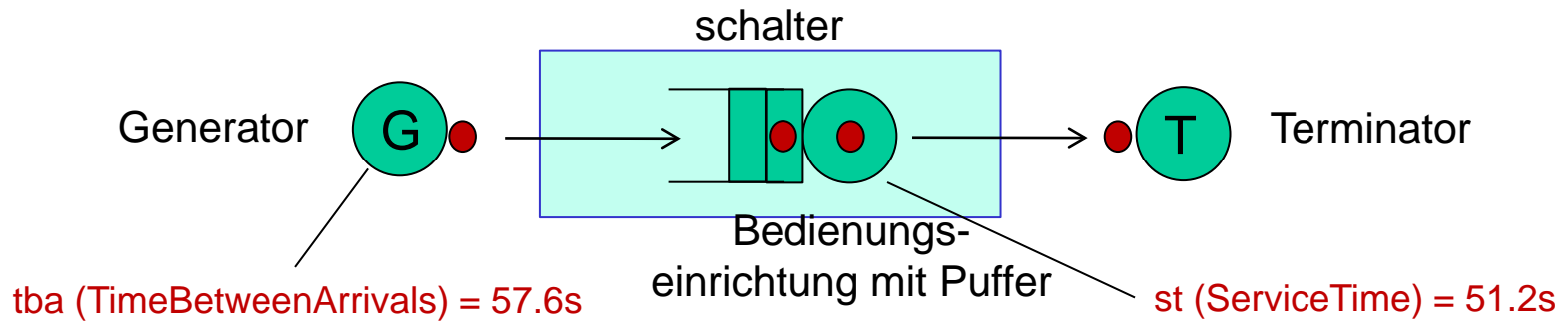


1. Zeit zwischen zwei aufeinanderfolgenden Transaktionserzeugungen



2. Verweildauer/Bedienungszeit einer Transaction in der Einrichtung

Bedienungssystem: Beispiel-1



GENERATE	tba ,,,1000
QUEUE	schalterQ
SEIZE	schalter
DEPART	schalterQ
ADVANCE	st
RELEASE	schalter
TERMINATE	1

*neuer Kunde wird nach tba Zeiteinheiten erzeugt
dieser zeigt dann eigenständiges Verhalten (1000 maximal)*

*aktueller Zeitpunkt des Kunden wird für Statistik **schalterQ** erfasst*

*Kunde belegt (falls möglich) zum aktuellen Zeitpunkt die Einrichtung **schalter**
sonst Blockierung (verlassender Kunde muss ihn wieder erwecken)*

*Wartezeitspanne des Kunden wird für Statistik **schalterQ** erfasst*

Kunde wird st Zeiteinheiten bedient

*Kunde verlässt zum aktuellen Zeitpunkt die Einrichtung **schalter** und
weckt den evtl. im Puffer von **schalter** wartenden Nachfolgekunden*

Reduziere einen Startzähler um eins

Objektorientierte Simulation mit ODEMX

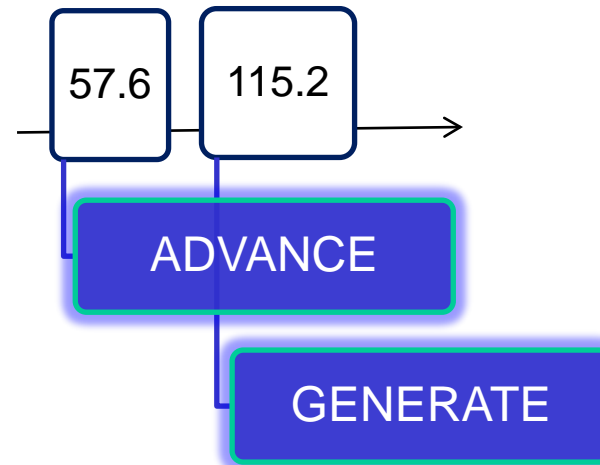
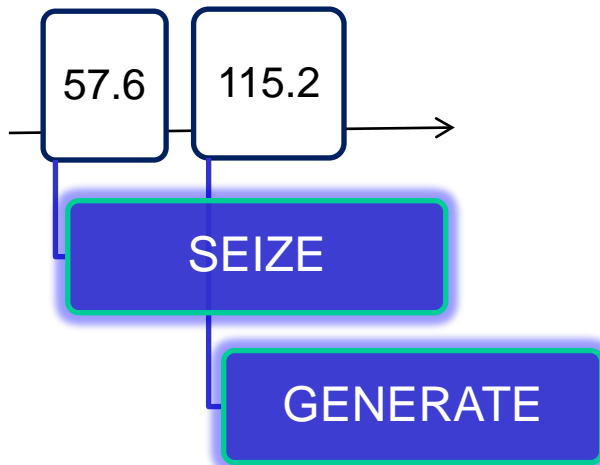
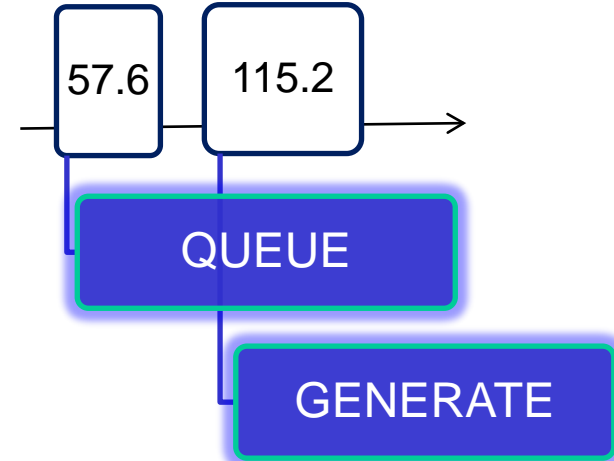
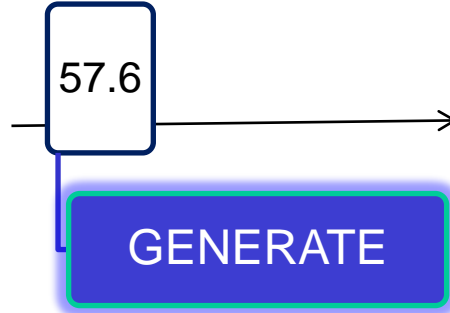
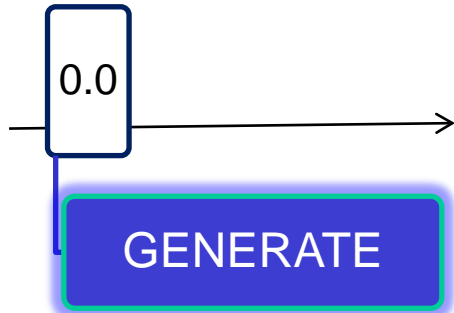
Grundphilosophie

- konzipiert für diskrete Ereignissimulation
- Bewegung von Transaktionen (Kunden, Produkte, Aufträge), die schrittweise bedient/ verarbeitet werden
- Bearbeitung/Bedienung erfolgt in Stationen (Blöcke)
- jede individuelle Transaction verändert in Abhängigkeit der aktuell belegten Einrichtung seine eigenen Attribute (Transactionzustand und die Attribute des Einrichtung und ...)
- bei Verbrauch von Modellzeit

1. Ereignis = Folge sequentieller Operationen zu einem fixierten Modellzeitpunkt
2. Ereignisse werden in einem Terminkalender chronologisch sortiert (man benötigt Lösung für Gleichzeitigkeitsprobleme) und in dieser Reihenfolge ausgeführt
3. aktuelle Modellzeit ist gleich der Ereigniszeit des ersten Kalendereintrages (aktives Element)
4. eine Ereignisoperation kann neben Zustandsänderungen neue Ereignisse planen (aber nur mit Ereigniszeiten, die größer oder gleich der aktuellen Modellzeit sind)

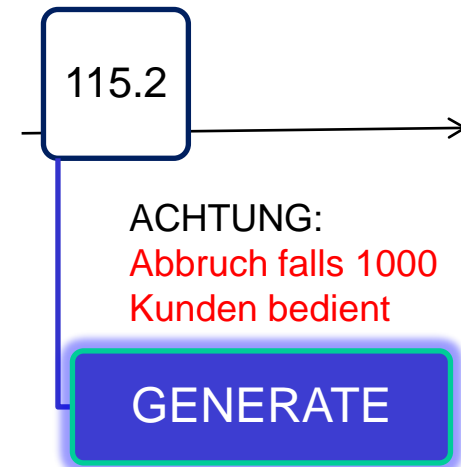
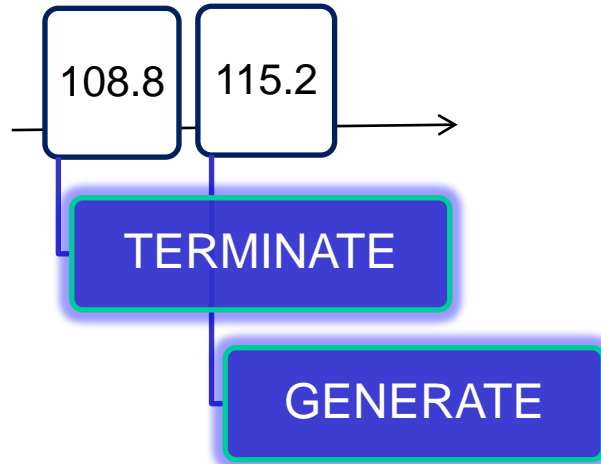
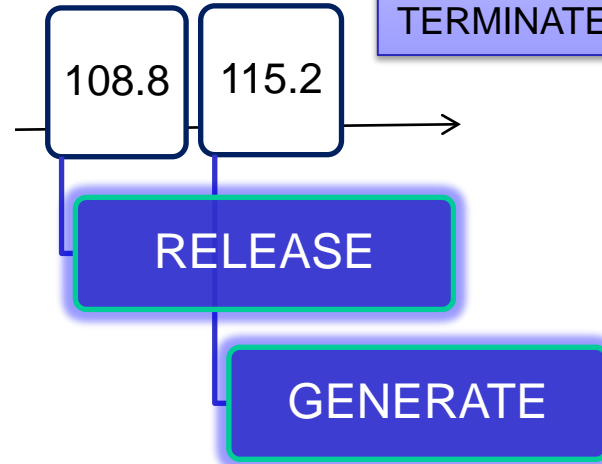
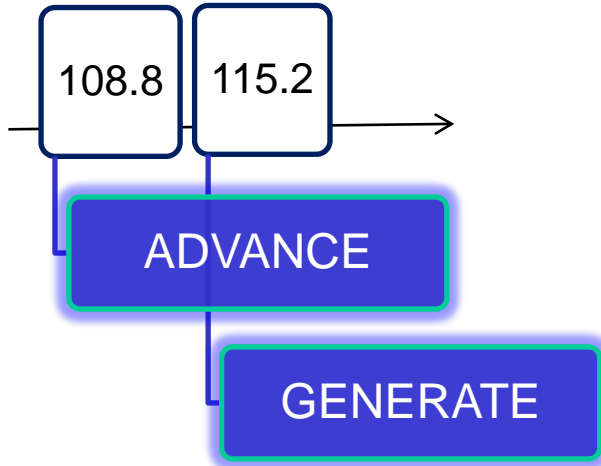
Ablauf als Ereignisfolge (1)

GENERATE	tba, 1000
QUEUE	schalterQ
SEIZE	schalter
DEPART	schalterQ
ADVANCE	st
RELEASE	schalter
TERMINATE	1



Ablauf als Ereignisfolge (2)

GENERATE	tba, 1000
QUEUE	schalterQ
SEIZE	schalter
DEPART	schalterQ
ADVANCE	st
RELEASE	schalter
TERMINATE	



Fazit (bereits aus OMSI-1 bekannt)

- Ereignis = mit einem Zeitpunkt (**Modellzeit**) verbundene Aktionsfolge (Zustandsänderung, Planung weiterer Ereignisse)
 - Ereignisoperation ist stets mit aktueller **Transaktion** und seiner aktuellen Bedienungsstation verbunden
 - Ereignisse sind chronologisch sortiert (**Ereigniskalender**)
 - Simulation = Ereignisrealisierung entsprechend der dynamisch stattfindenden Planung im Terminkalender
 - man beginnt mit der Ausführung des ersten Ereignisses und
 - geht dann zum nächsten Ereignis
- Methode: **Next-Event-Simulation**
- Ende der Simulation (falls Ereigniskalender leer, oder explizite Beendigungsoperation als Ereignis)
 - Systemverhalten bei festen Zeiten (Ankunfts- und Bedienungszeiten) relativ einfach, aber nicht bei variierenden (zufälligen) Zeitangaben!

Standardausgabe eines GPSS-Simulators

Für alle Bedienungseinrichtungen: Statistik

FACILITY	ENTRIES	UTIL.	AVE._TIME	AVAILABLE	OWNER
schalter	1000	0.879	50.98	1	0

totale Anzahl von Eintritten (links auf ENTRIES)
 Auslastung (links auf UTIL.)
 mittlere Belegung (links auf AVE._TIME)
 augenblickliche Verfügbarkeit (links auf AVAILABLE)
 augenblickliche Belegung (links auf OWNER)

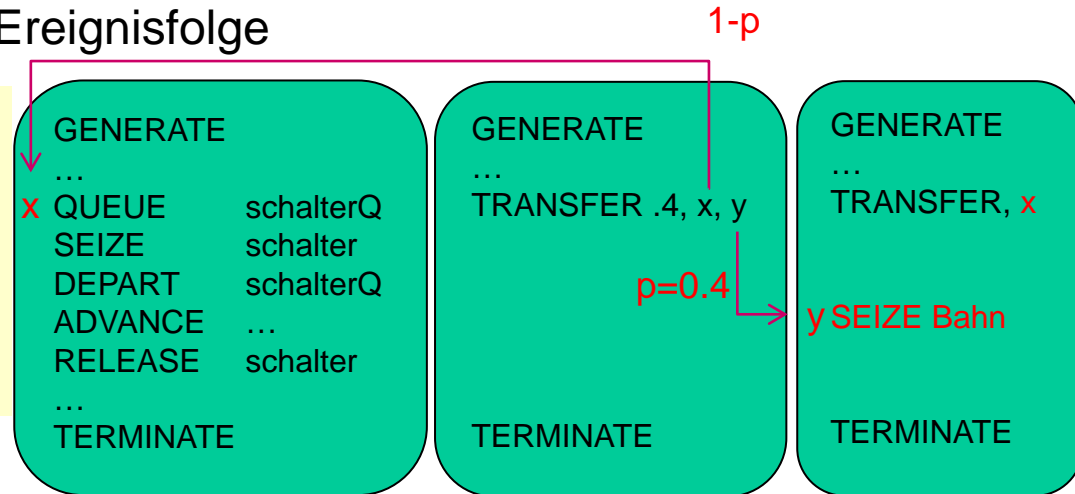
QUEUE	MAX	CONT.	ENTRIES	ENTRIES(0)	AVE.CONT.	AV.TIME	AVE.(-0)
schalterQ	1	0	1000	895	0.00	0.22	2.14

maximale Länge (links auf MAX)
 augenblickliche Länge (links auf CONT.)
 totale Anzahl von Eintritten (links auf ENTRIES)
 Gesamtanzahl von Eintritten ohne Durchläufer (d.h. mit null Wartezeit) (links auf ENTRIES(0))
 mittlere Anzahl von Eintritten (links auf AVE.CONT.)
 mittlere Wartezeit (aller Belegungen) (links auf AV.TIME)
 mittlere Wartezeit der echt wartenden Belegungen (links auf AVE.(-0))

Aktive und stationäre Modelleinheiten (2)

- Transaktion (=Prozess-Instanz)
Objekt + strukturierte Ereignisfolge

*Transaktionen bewegen sich von Station (Block) zu Station mit evtl. Zeitverzögerung
→ geteilte Nutzung von Ressourcen durch konkurrierende Transaktionen*



Transaktionen sind nicht typisiert

müssen vom Modellierer durch Merkmale/Nummern unterschieden werden

alle haben die gleiche Datenstruktur:

- Attribute mit (fester) Built-in-Semantik
- Attribute mit (flexibler) nutzer-spezifischer Semantik

- Stationen sind stationär (statisch: Vorgänger, Nachfolger)

Bewegung der Transaktionen

*können Verzweigungen/Sprünge (TRANSFER) zu anderen Blöcken sein
(globale Namensgebung)*

Transaktion (Attribute)

Zugriff	Bedeutung
-Laufzeitsystem-	Nummer der Transaktion
MC	Erzeugungszeitpunkt
PR	Prioritätswert
-Laufzeitsystem-	Nummer der zeitlich vorangehenden Transaktion
-Laufzeitsystem-	Nummer der zeitlich nachfolgenden Transaktion
-Laufzeitsystem-	Nummer des augenblicklichen Blockes
-Laufzeitsystem-	Blockabgangszeit
P1	Parameter 1
...	
Pn	Parameter n

eigene Namenswahl möglich

Zugriff P\$ParameterName oder P*nr*

Kommunikation von Transaktionen

- Nur über globale Größen (Zugänge über Makros) mit vorheriger selbst zu organisierender Synchronisation (eingeschränkte Möglichkeiten)
 - SNA-Größen (System Numeric Attribute) (z.B. Zustandsinformationen von Stationen, und des Simulationslaufzeitsystem)
 - Skalare (Modellkonstanten, Modellvariablen)
 - Felder

Int
Float
String
Boolean

Warteschlangen-spezifische SNAs

Zugriff	Bedeutung
-Laufzeitsystem-	Nummer oder Name j der Warteschlange
QC j	Anzahl der bisherigen Eintritte
Q j	momentane Länge
QM j	maximale Länge
QA j	mittlere Länge
QT j	mittlere Wartezeit der Transaktionen

QM\$schalterQ



GENERATE	tba, 1000
QUEUE	schalterQ
SEIZE	schalter
DEPART	schalterQ
ADVANCE	st
RELEASE	schalter
TERMINATE	1

```

; GPSS World Sample File - SAMPLE9.GPS Model to demo graphics windows
Pool      STORAGE 400          ;Define Storage
Matrix1   MATRIX ,5,5         ;Define Matrix
Transit   TABLE M1,200,200,20 ;Transit time in wait ; and process

        GENERATE (Exponential(1,0,100))
        JOIN      Maingrp      ;Xact joins grp called ; Maingrp
        JOIN      Numgrp,9999  ;Add 9999 to Numeric
        SAVEVALUE Addup+,1     ;Total of Transactions
        ASSIGN    Param_1,232  ;Assign Xact parameter
        JOIN      Numgrp,P$Param_1 ;Put value in Param1
        LOGIC S    Switch_1    ;Turn on a logic switch
        MSAVEVALUE Matrix1,2,2,QA$Tot_Process
        QUEUE     Tot_Process  ;Queue for process time
        SEIZE     Facility1    ;Own first Facility
        LINK      Chain1,FIFO,Nxtblk ;Put on Userchain if busy
Nxtblk   SEIZE     Facility2    ;Own a second Facility
        SEIZE     Facility3    ;Own a third Facility
        QUEUE     Process_Time ;Keep track of process
        ADVANCE   100,(Exponential(1,0,100))
        DEPART    Process_Time ;Record length
        TABULATE  Transit      ;Add wait + process
        RELEASE   Facility1    ;Give up 1st Facility
        ADVANCE   20           ;Delay time for Fac 2&3
        RELEASE   Facility2    ;Give up 2nd Facility
        ADVANCE   10           ;Extra delay time-Fac 3
        RELEASE   Facility3    ;Give up 3rd Facility
        DEPART    Tot_Process  ;Leave Queue
        UNLINK    Chain1,Nxtblk ;Take off waiting Xacts
        ENTER     Pool,100     ;Place 100 units in the Storage
        LOGIC R    Switch_1    ;Turn off logic switch
        LEAVE     Pool,50      ;Take 50 units from Storage
        SAVEVALUE Collect-,1  ;Show negative Savevalue
        REMOVE    Maingrp     ;Remove Xact from group
        ADVANCE   50,1        ;Wait 50(+/-1) time units
        LEAVE     Pool,50      ;Take 50 units from Storage
Finis    TERMINATE 1          ;Destroy Xact

```

*nur die Stationen
sind zu deklarieren,
die Parameterangaben
für ihre Konstruktion
benötigen*

5. GPSS

1. Grundphilosophie

2. Block/Stations-Übersicht

Transaktionsverwaltung (1)

zwei interne Ereignislisten

- **CurrentEventChain**: Aktuelle Ereignisliste (CEC)
- **FutureEventChain**: Zukünftige Ereignisliste (FEC)

ODEMx-
Terminkalender

CEC

- verwaltet alle aktiven Transaktionen zum aktuellen Modellzeitpunkt
als priorisierte Listeneinträge

unterschiedliche Handhabung der blockierten Transaktionen
in verschiedenen GPSS-Versionen
(manchmal in separaten Nutzer-Listen:
DelayChain, **RetryChain**, ...)

FEC

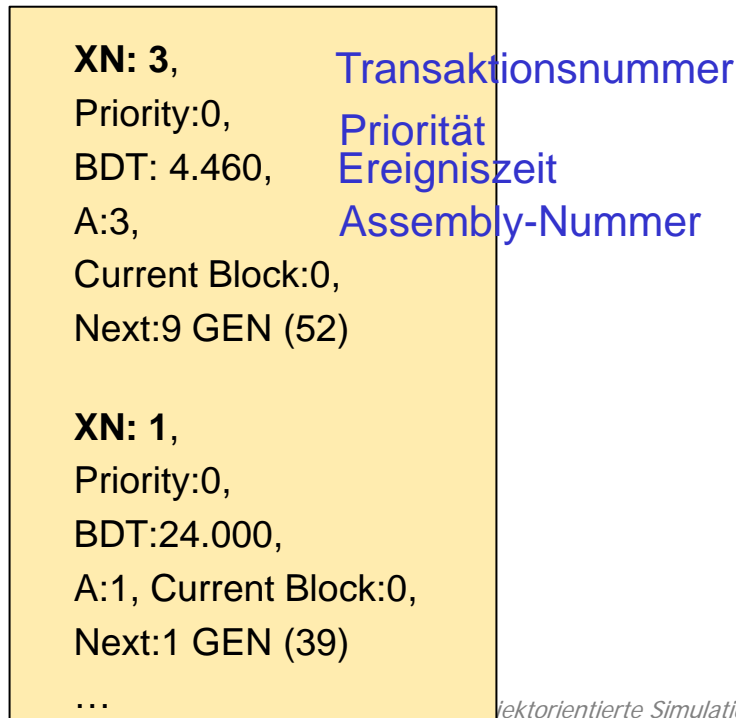
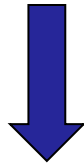
- alle aktiven Transaktionen zu späteren Modellzeitpunkten
(sortiert als priorisierte Liste)

nur **GENERATE** und **ADVANCE** erzeugen Ereignisse in FEC

Transaktionsverwaltung (2)

Beobachtung:

window > simulation snapshot >
CEC/FEC snapshot



- CEC
enthält stets Ereignisse zum aktuellen Zeitpunkt:
priorisierte/ sequentielle
Realisierung aller Ereignisse
(bei evtl. Generierung neuer
Ereignisse)

falls CEC leer:
Erhöhung der aktuellen Zeit,
bei Übernahme der
entsprechenden Ereignisse
aus FEC

- FEC,
chronologisch sortierte Liste

Grundzustände einer Transaktion

Zustand = Grundzustand × Attribute (Standard, nutzerdef)

- **ACTIVE**

The Transaction is the highest priority Transaction in the CEC.

- **SUSPENDED**

The Transaction is waiting in the FEC or the CEC to become the active Transaction.

- **PASSIVE**

The Transaction has come to rest in the simulation on a [User Chain](#), [Delay Chain](#), [Retry Chain](#), [Interrupt Chain](#), or [Pending Chain](#).

- **TERMINATED**

The Transaction has been destroyed and no longer exists in the simulation.

Ablauf- verfolgung

Menu command:

window >

simulation window >

blocks window

Loc	Block Type	Current Count	Entry Count	Retry Chain	Line Number	Include-f...
1 GEN	GENERATE	0	81	0	39	0
2 TER	TERMINATE	0	81	0	40	0
3 GEN	GENERATE	0	1	0	44	0
AGAIN	LOGIC	0	151	0	45	0
5 ADV	ADVANCE	1	151	0	47	0
6 LOG	LOGIC	0	150	0	48	0
7 ADV	ADVANCE	0	150	0	49	0
8 TRA	TRANSFER	0	150	0	50	0
9 GEN	GENERATE	1	75	0	52	0
10 TRA	TRANSFER	0	74	0	53	0
11 ASN	ASSIGN	0	40	0	56	0
12 ASN	ASSIGN	0	40	0	57	0
13 ASN	ASSIGN	0	40	0	59	0
14 ASN	ASSIGN	0	40	0	61	0
15 ASN	ASSIGN	0	40	0	62	0
16 ASN	ASSIGN	0	40	0	63	0
17 QUE	QUEUE	0	40	0	64	0
18 TRA	TRANSFER	0	40	0	65	0
PIER1	GATE	0	27	0	67	0
20 ASN	ASSIGN	0	27	0	68	0
SMALL	ENTER	0	74	0	70	0
22 DEP	DEPART	0	74	0	72	0
23 ADV	ADVANCE	0	74	0	73	0
24 ADV	ADVANCE	1	74	0	74	0
25 TES	TEST	0	73	0	75	0
26 GAT	GATE	0	12	0	77	0
SKIPIT	LEAVE	0	73	0	78	0
28 TAB	TABULATE	0	73	0	80	0
29 TER	TERMINATE	0	73	0	82	0
PIER2	TEST	0	13	0	86	0
31 TRA	TRANSFER	0	13	0	87	0
BERT2	TEST	0	13	0	88	0
33 ASN	ASSIGN	0	13	0	89	0
34 TRA	TRANSFER	0	13	0	90	0
BERT3	TEST	0	0	0	91	0
36 ASN	ASSIGN	0	0	0	92	0
37 TRA	TRANSFER	0	0	0	93	0
INTER	TRANSFER	0	34	0	96	0
39 PRI	PRIORITY	0	22	0	98	0
40 ASN	ASSIGN	0	22	0	100	0
41 ASN	ASSIGN	0	22	0	102	0
42 ASN	ASSIGN	0	22	0	104	0
43 ASN	ASSIGN	0	22	0	105	0
44 ASN	ASSIGN	0	22	0	106	0
45 ASN	ASSIGN	0	22	0	107	0

Obj