

Modul OMSI-2 ***im SoSe 2011***

Objektorientierte Simulation ***mit ODEMx***

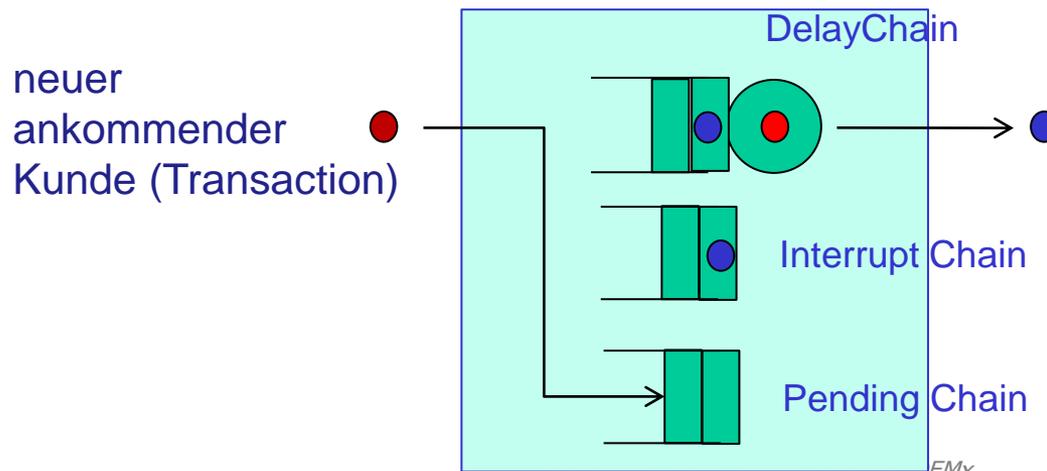
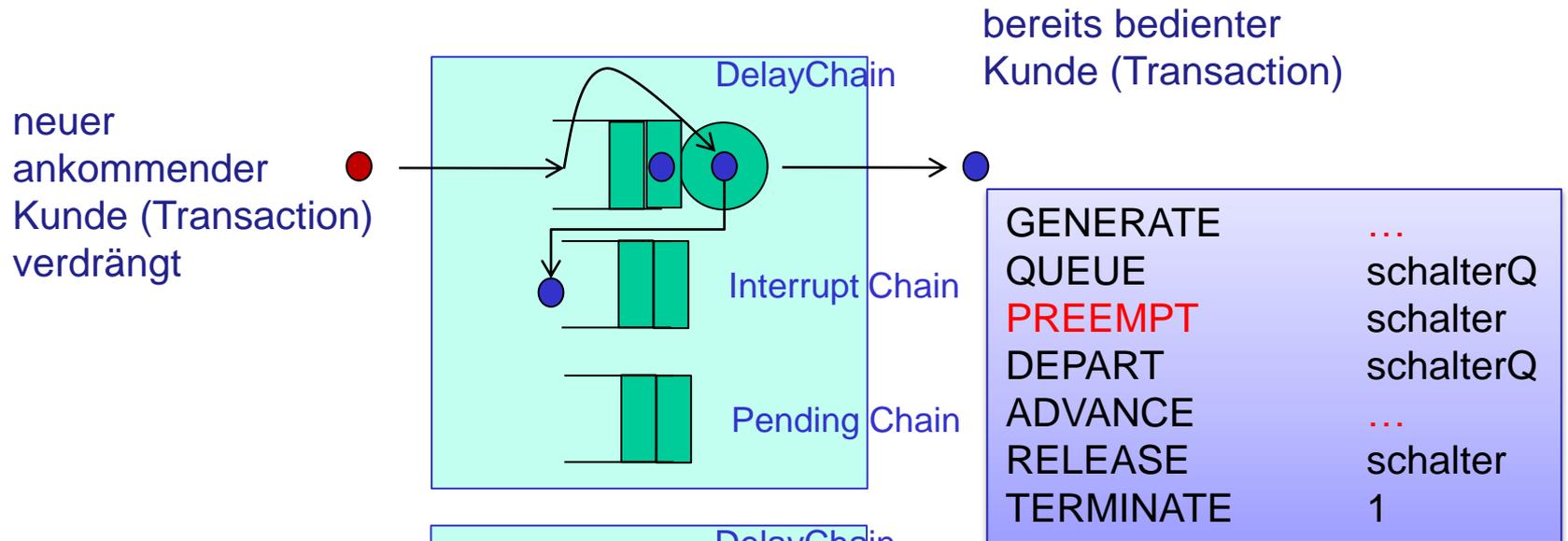
Prof. Dr. Joachim Fischer
Dr. Klaus Ahrens
Dipl.-Inf. Ingmar Eveslage
Dipl.-Inf. Andreas Blunk

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

5. GPSS

1. Grundphilosophie
2. Block/Stations-Übersicht (1)
3. Einfaches GPSS-Beispiel
4. Erste Einschätzung der GPSS-Ausdruckskraft
5. Vordefinierte Modellierungsmuster (FACILITY)
6. Standardattribute
7. Block/Stations-Übersicht (2)
8. Nutzereigene Modellierungsmuster (asynchrone Kommunikation)

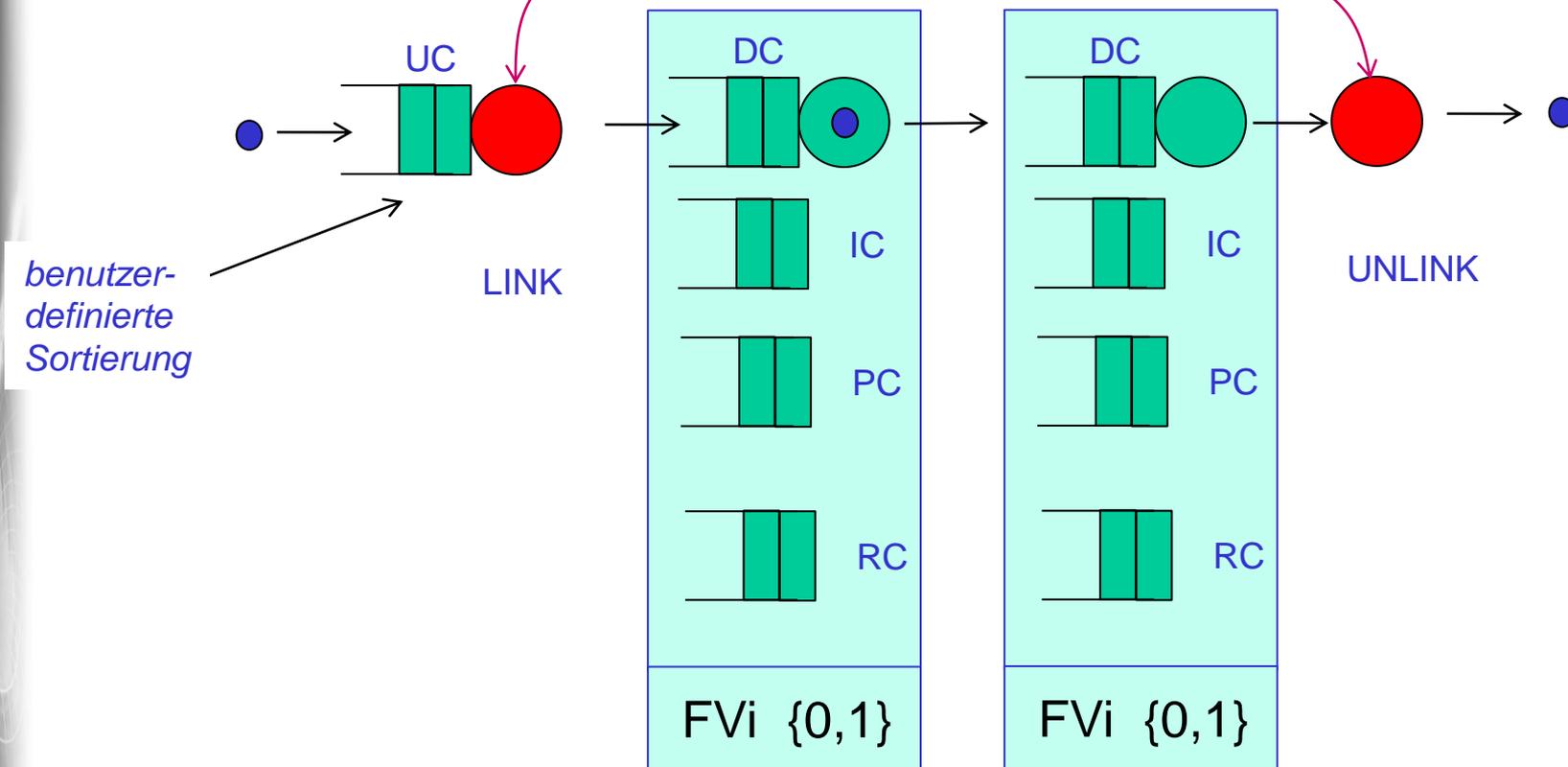
PREEMPT: Einfache Szenarien



Objektorientierte Simulation mit QEMX

Nutzerketten, User Chains (UC)

als Paar



Zu *Transaktionsattributen*

1. Parameter (nutzereigene Attribute)
müssen erzeugt und initialisiert sein- bevor auf sie zugegriffen werden darf

Dies wird erreicht mit:

- **ASSIGN**,
- **MARK**,
- TRANSFER SUB,
- SELECT,
- **SPLIT**,
- COUNT

Zu Transaktionsattributen

2. Priorität

wird benutzt bei Blockierungwarteschlangen

insbesondere bei:

- Current Events Chain,
- Facility Delay Chains,
- Storage Delay Chains

FRAGE: Kann man die Priorität dynamisch ändern?

ANTWORT: JA,

PRIORITY Wert

Zu Transaktionsattributen

3. MarkTime

Modellzeitpunkt,
zu dem die Transaktion im System zum ersten Mal auftritt
(GENERATE-Zeitpunkt)

oder

der Zeitpunkt des Betretens des letzten MARK- Blockes
(ohne Parameterangabe)

Anwendung: Benutzung des vordefinierten Funktionen/Makros
M1 zur Bestimmung der bisherigen Laufzeit einer
Transaktion

M1 := C1 – MarkTime der aktuellen Transition
(C1 = aktuelle Modellzeit)

Zu Transaktionsattributen

4. AssemblySet

- positiver Integer-Wert
(Codierung einer Gruppenzugehörigkeit von Transaktionen)
- AssemblySets zur Transaktionenssynchronisation
 - ASSEMBLE,
 - GATHER und
 - MATCH Blocks
- Initialwert:
 - per GENERATE ist gleich der Transaktionsnummer
 - per SPLIT ist gleich dem AssemblySet-Wert der Erzeuger-Transaktion
- Änderung durch ADOPT-Block

ADOPT 2000

Zu Transaktionsattributen

5. StatusFlags

- Delay Indicator
- Trace indicator (Blöcke: TRACE, UNTRACE)

6. Position, Fluss

- currentBlock
- nextBlock

7. Grundzustand

- ACTIVE - höchste Priorität in der CEC.
- SUSPENDED – wartet in FEC oder CEC
- PASSIVE - blockiert in UserChain, DelayChain, PendingChain oder RetryChain.
- TERMINATED - nicht mehr aktivierbar

zusätzlich

- PREEMPTED - per PREEMPT verdrängt, wartet in InterruptChain

sind dem Nutzer nicht zugänglich?

5. GPSS

1. Grundphilosophie
2. Block/Stations-Übersicht (1)
3. Einfaches GPSS-Beispiel
4. Erste Einschätzung der GPSS-Ausdruckskraft
5. Vordefinierte Modellierungsmuster (FACILITY)
6. Standardattribute
7. Block/Stations-Übersicht (2)
8. Nutzereigene Modellierungsmuster (asynchrone Kommunikation)

Erzeugung von Transaktionen

GENERATE a, b, c, d, e, f, g, h, i

ist ein Block zur Erzeugung von Transaktionen
nach der Erzeugung durchlaufen diese die nachfolgenden Blöcke
bis Erreichen eines TERMINATE-Blockes

- Erzeugungsabstände: gleichverteilt im Intervall [a-b, a+b]
- Operand b kann leer sein, dann äquidistante Erzeugungsabstände (oder Zufallsgeneratorangabe)
- Operand c ist der Zeitpunkt des ersten Aktivators, der GENERATE verlässt (kann leer sein)
- Operand d ist maximale Anzahl von Aktivatoren (leer bedeutet unbeschränkt)
- Operand e ist Prioritätswert [0, 127] (leer Null)
- Operand f bis i, nutzerspezifische Parameter (haben den Anfangswert Null)

Wertzuweisungen für Transaktionsparameter

ASSIGN $a(+,-)$, b , c

ist ein Block zur Wertzuweisung für Transaktionsparameter

Zuweisung erfolgt bei jedem Betreten des Blockes durch eine Transaktion

- Operand a ist Nummer/Name des Parameters, der aktualisiert werden soll
- Operand $a+$ (Wert wird addiert), $a-$ (Wert wird subtrahiert)
- Operand b ist der Wert / Summand, Subtrahend
- Operand c ist Nummer/Name einer Funktion

ASSIGN 2000,150.6

ASSIGN Text,"Look on my works."

ASSIGN 2000-, -3

P2000

P\$Text

Zugriff

Parametername wird angelegt, falls noch nicht vorhanden

Typen: String, Integer, Float, Boolean

Wertzuweisungen an globale Größen

SAVEVALUE a, b

ist ein Block zur Wertzuweisung an einfache globale Variablen
Zuweisung erfolgt bei jedem Betreten des Blockes durch eine Transaktion

- Operand **a** ist Nummer/Name des Skalars
- Operand **b** ist der zuzuweisende Wert

SAVEVALUE Account, 99.95

X\$Account

*globale Variablen besitzen Retry-Chain
enthalten Transaktionen, die zustandsbedingt im TEST-, ...Block blockiert sind*

Feld-Operationen

Label MATRIX a, b, c

ist eine Anweisung zur Definition eines Feldes **Label**

- **a** wird nicht benutzt
- **b** Zeilenzahl
- **c** Spaltenzahl

MSAVEVALUE a(+/-), b: Row, c: Colum, d:Value

- ist ein Block zur Wertzuweisung für ein Matrixfeld
- **a** ist Name/Nummer der Matrix

INITIAL a, b

- ist eine Anweisung zur Definition eines Feldes, Skalaren Größe, Logischer Schalter
- **a** ist Name/Nummer der Matrix
- **b** Wert

```
INITIAL MX$feld(3,2), 17
```

```
INITIAL X$var, 1500
```

```
INITIAL LS$switch, true
```

Vervielfachung von Transaktionen

SPLIT a, b, c

ist ein Block, der Kopien der ihn betreffenden Transaktion erzeugt

- Operand **a** ist Anzahl der zu erzeugenden Kopien
- Operand **b** ist Nummer oder Name des Blockes, zu dem die Kopien bewegt werden sollen,

Das Original wird zum Folgeblock bewegt

(Ist **b** leer werden auch die Kopien zum Folgeblock bewegt)

- Operand **c** gibt Nummer des Parameters, der zur lfd. Nummerierung von Original und Kopien benutzt wird

SPLIT 3, Pro, 17

SPLIT 3, Pro, P\$17

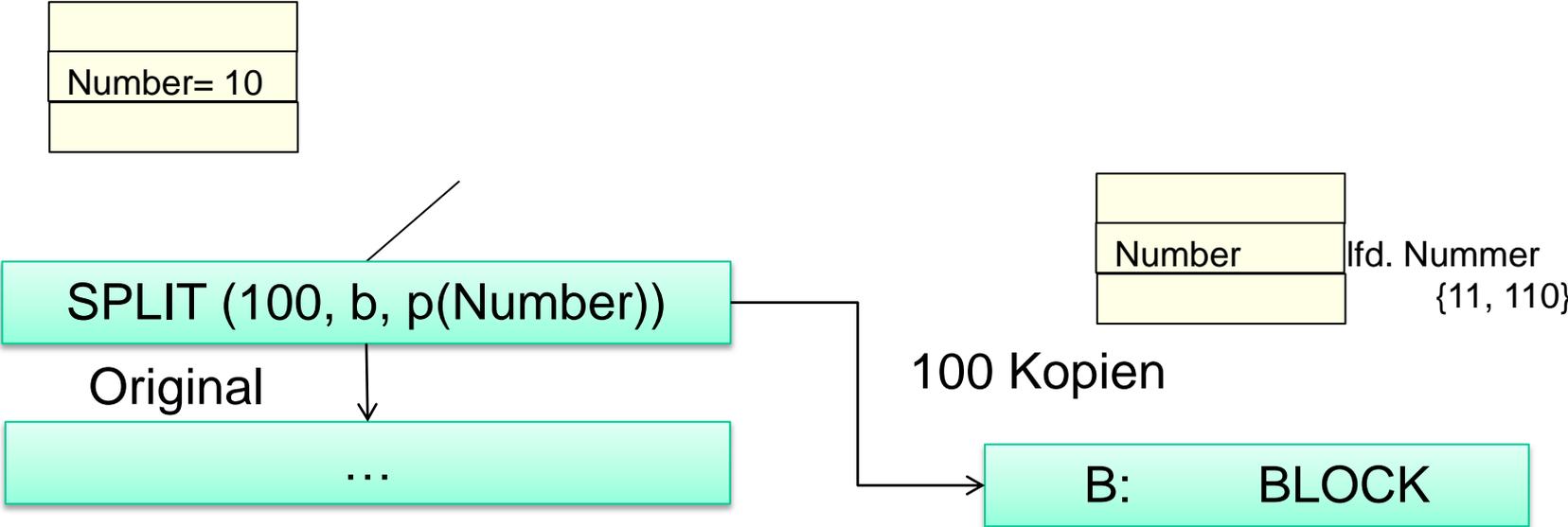
Unterschied klarmachen !

Auszug Ref.Manual:

each time the split Block is entered, 3 new Transactions are created. The parent Transaction goes to the Next Sequential Block, the offspring to a Block labeled Pro. Parameter 17 will receive the serialization.

If the parent Transaction's Parameter is not predefined, it will be created and initialized to 0. In this example, the parent Transaction with a value of 0 in the Parameter 17, (the Parameter to be used for serialization) will, after passing through the SPLIT Block, have a 1 in Parameter 17 and the offspring will have 2, 3, and 4 in Parameter 17

Nummernverwaltung



Zeitmarkierung

MARK a

ist ein Block zur Aktualisierung von Zeitmarkierungen

- Operand **a** ist ein Parameter, zur Speicherung der aktuellen Modellzeit (nutzereigene Speicherung von Zeitstempeln), kein Einfluss auf **M1**, aber auf **MP**)

Wird kein Parameter angegeben, wird die aktuelle Modellzeit im Attribut MarkTime gespeichert

MARK

...

TEST G M1,70000

C1 – MarkTime der aktuellen Transaktion

MARK Startzeitpunkt

...

TEST G MP\$Startzeitpunkt,70000

C1 – Startzeitpunkt der aktuellen Transaktion

Warteschlangenstatistik

QUEUE a, b

ist ein Block zur Erfassung der zeitlichen Verzögerung von Transaktionen

Operand **a** ist Name/Nummer der Warteschlange, die betreten wird

- Operand **b** ist Inkrement-Wert (optional) für Queue-Inhalt

DEPART a, b

ist ein Block zur statistischen Erfassung von Transaktionen
(in einer (Warteschlange))

- Operand **a** ist Name/Nummer der Warteschlange, die verlassen wird
- Operand **b** ist Dekrement-Wert (optional) für Queue-Inhalt

Verzögerung

ADVANCE a, b

- ist ein Block zur statistischen Erfassung von Transaktionen (in einer Warteschlange)
- Verzögerungswert: gleichverteilt im Intervall [$a-b$, $a+b$]

```
ADVANCE 3
```

```
...
```

```
ADVANCE 90, 5
```

```
...
```

```
ADVANCE 120, FN$Exp
```

```
...
```

```
ADVANCE V$Norm
```

Vernichtung von Transaktionen

TERMINATE **a**

ist ein Block zur Vernichtung von Transaktionen

- Operand **a** gibt Anzahl von Einheiten an, um die der Wert des Startzählers verringert wird (Count-Down-Zähler)
- erreicht der Startzähler Null, bricht die Simulation ab

START **A, B**

- ist eine Steueranweisung, die die Ausführung einer Simulationsphase steuert
- Operand **a** ist der Anfangswert des Startzählers
- Operand **B** steuert die Ausgabe des Simulators (leer: Standardausgabe)

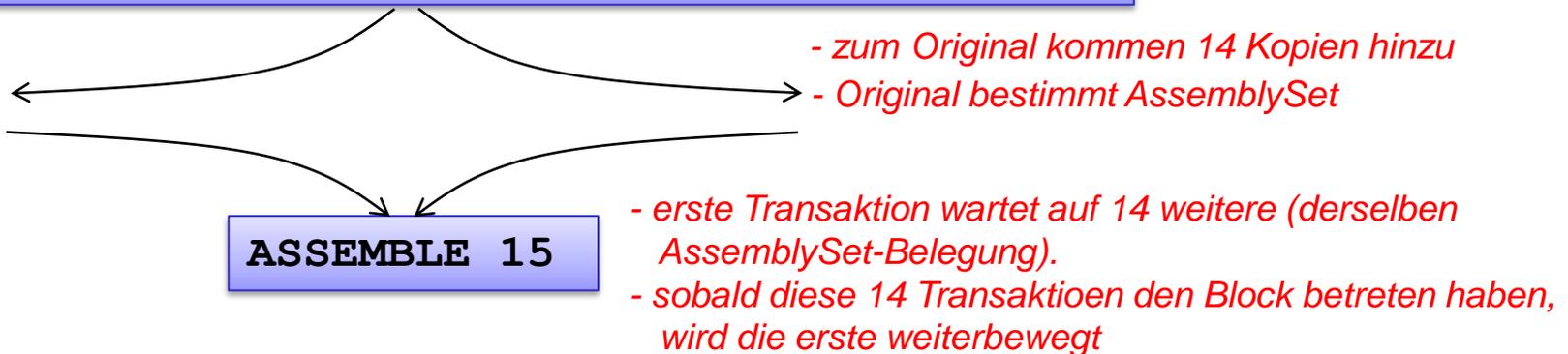
Behandlung von Transaktionsgruppen

ASSEMBLE *a*

ist ein Block zur Synchronisation und Zusammenlegung von *n* Transaktionen der gleichen AssemblySet-Belegung bei Vernichtung der „überflüssigen“ Transaktionen

- Operand *a* ist der Transaktionszähler *n* (≥ 0)

SPLIT 14, CopyDestination, P\$SerialNumber



Behandlung von Transaktionsgruppen (Forts.)

GATHER **a**

ist ein Block zur Synchronisation von **n** Transaktionen der gleichen AssemblySet-Belegung bei Fortsetzung aller Transaktionen

- Operand **a** ist der Transaktionszähler **n** (≥ 0)

SPLIT 14, CopyDestination, P\$SerialNumber



GATHER 15

- erste Transaktion wartet auf 14 weitere (derselben AssemblySet-Belegung).
- sobald diese 14 Transaktionen den Block betreten haben, werden alle gleichzeitig(sequentiell) weiterbewegt

Behandlung von Transaktionsgruppen (Forts.)

MATCH **a**

ein Paar von MATCH-Blöcken dient der Synchronisation von **einem** Transaktionspaar der gleichen AssemblySet-Belegung bei Fortsetzung beider Transaktionen

- Operand **a** ist die Identifikation des konjugierten MATCH-Blockes

SPLIT 14, CopyDestination, P\$SerialNumber

X MATCH Y

Y MATCH X

- zum Original kommen 14 Kopien hinzu
- Original bestimmt AssemblySet

Verzweigung und Vereinigung von Transaktionsströmen (1)

TRANSFER A, B, C, D

- ist ein Block zur Verzweigung von Transaktionsströmen bzw. zur Vereinigung von zwei Strömen
- Nächste Block:=
 - a) Operand **A** ist leer,
B ist dann Nummer/Name des Folgeblockes
 - b) Operand **A** enthält eine Wahrscheinlichkeitsangabe p (.xxx)
B ist dann Nummer/NAME des Blockes zu dem mit Wahrscheinlichkeit $1-p$ verzweigt wird
(**B** leer, wird zum Folgeblock verzweigt)
Operand **C** gibt Nummer/Name des Blockes an, zu dem die Transaktion mit Wahrscheinlichkeit p verzweigt wird
 - c) Operand A ist Modus: ALL ...?

ZielBlock

*verwaltet implizit Kette gemischter Transaktionen,
sortiert nach Priorität und FIFO bei Gleichheit
(Verzögerungskette/RetryChain)*

Sprung

BOTH, ALL, PICK, FN, P, SBR, SIM,
Ausdruck, SNA

label TRANSFER a, b, c, d

springt zu einem bestimmten Block.

a – Wahrscheinlichkeit oder Mode

b – Block-Angabe

c – Block-Angabe

d – Inkrement für den Fall a=ALL (Standardwert 1)

label TRANSFER , CLERK1

unbedingter Sprung

...

TRANSFER BOTH, T1, T2

*zum ersten verfügbaren
Block entweder T1 oder T2*

*Die blockierte Transaktion wird in den RetryChains der
Verfügbarkeitsattribute der beiden Blöcke vermerkt:*

- SEi (Speicherverfügbarkeit)
- FVi (Einrichtungsverfügbarkeit)

TRANSFER: Unconditional Mode

TRANSFER ,New_Place

When a Transaction enters this TRANSFER Block,
it is immediately scheduled for the Block at location New_Place

Ein Sprung zu einem GENERATE-Block führt zum Abbruch der Simulation mit Fehlermeldung

TRANSFER: Fractional Mode

TRANSFER .75 , ,New_Place

When a Transaction enters this TRANSFER Block, it proceeds to the location named NEW_PLACE with a probability of .75.

The remaining times it proceeds to the Next Sequential Block. You can select which random number generator number is to be used as the source of the random number.

This is set in the "Random" page of the Model Settings Notebook.

TRANSFER: Both Mode

TRANSFER BOTH, First_Place, Second_Place

When a Transaction enters this TRANSFER Block, the Block at location First_Place is tested. If the Transaction can enter, it does so. If not, the Block at location Second_Place is tested. The Transaction enters if it can. Otherwise, it remains in the TRANSFER Block until it can leave

Die blockierte Transaktion wird in den RetryChains der Verfügbarkeitsattribute der beiden Blöcke vermerkt:

- SEi (Speicherverfügbarkeit)
- FVi (Einrichtungsvfügbarkeit)

andere Blöcke (bis auf GENERATE) können unproblematisch angesprungen werden

TRANSFER: All Mode

TRANSFER **ALL**, First_Place, Last_Place, 2

When a Transaction enters this TRANSFER Block, the Block at location First_Place is tested. If the Transaction can enter, it does so.

If not, the Blocks at every second higher location are tested. The Transaction enters if it can.

If all tested Blocks refuse, the testing ends with the Block at location Last_Place, or with the Block just before it, depending on the separation of First_Place and Last_Place.

If no Block accepts, the Transaction remains in the TRANSFER Block until it can leave

Die blockierte Transaktion wird in den RetryChains der Verfügbarkeitsattribute der aller Blöcke vermerkt:

- SEi (Speicherverfügbarkeit)
- FVi (Einrichtungsvfügbarkeit)

andere Blöcke (bis auf GENERATE) können unproblematisch angesprungen werden

TRANSFER: Pick Mode

TRANSFER **PICK**, First_Place, Last_Place

When a Transaction enters this TRANSFER Block, a location is chosen randomly which is numerically between First_Place and Last_Place, inclusively. The chosen location is the next destination for the Active Transaction. You can select which random number generator number is to be used as the source of the random number.

Sollte die Transaktion blockieren, wird sie in den RetryChains der Verfügbarkeitsattribute des gewählten Blockes vermerkt:

- SEi (Speicherverfügbarkeit)
- FVi (Einrichtungsvfügbarkeit)

TRANSFER: Function Mode

TRANSFER **FN**, Func1, 5

When a Transaction enters this TRANSFER Block, the function entity named FUNC1 is evaluated, and added to 5, to determine the location of the destination

Sollte die Transaktion blockieren, wird sie in den RetryChains der Verfügbarkeitsattribute des gewählten Blockes vermerkt:

- SEi (Speicherverfügbarkeit)
- FVi (Einrichtungsvfügbarkeit)

TRANSFER: Parameter Mode

TRANSFER **P**, Placemarker, 1

When a Transaction enters this TRANSFER Block, it is immediately scheduled for the Block immediately after the location specified in the Transaction Parameter named Placemarker.

Sollte die Transaktion blockieren, wird sie in den RetryChains der Verfügbarkeitsattribute des gewählten Blockes vermerkt:

- SEi (Speicherverfügbarkeit)
- FVi (Einrichtungsvfügbarkeit)

TRANSFER: Subroutine Mode

TRANSFER **SBR**, New_Place, Placemark

When a Transaction enters this TRANSFER Block, it is immediately scheduled for the Block at location New_Place. The location of the TRANSFER Block is placed in the Parameter named Placemark. If there is no such Parameter, it is created.

To return from the subroutine, use a TRANSFER Block in Parameter Mode.

Sollte die Transaktion blockieren, wird sie in den RetryChains der Verfügbarkeitsattribute des gewählten Blockes vermerkt:

- SEi (Speicherverfügbarkeit)
- FVi (Einrichtungsvfügbarkeit)

TRANSFER: Simultaneous Mode

TRANSFER **SIM**, Nodelay_Place, Delay_Place

When a Transaction enters this TRANSFER Block, it is immediately scheduled for the Block at location New_Place. The location of the TRANSFER Block is placed in the Parameter named Placemark. If there is no such Parameter, it is created.

To return from the subroutine, use a TRANSFER Block in Parameter Mode.

Sollte die Transaktion blockieren, wird sie in den RetryChains der Verfügbarkeitsattribute des gewählten Blockes vermerkt:

- SEi (Speicherverfügbarkeit)
- FVi (Einrichtungsvfügbarkeit)

SELECT

FNV, FV, I, LS, LR, NI, NU, SE, SF, SNE, SNF, SNV, SV,
U, E, G, GE, LE, MIN, MAX, NE

label SELECT co a, b, c, d, e

bestimmt eine Entität, die der Bedingung entspricht,
und legt Identifikation in einen Parameter der aktiven Transition ab.

- a – Angabe des Parameters
- b – untere Entitätsnummer
- c – obere Entitätsnummer
- d -
- e – SNA-Klassenname

label SELECT MIN 4, CLERK1, CLEARK7, , Q
 QUEUE P4
 SEIZE P4

Auswahl der Einrichtung mit kürzester Warteschlange

Zustandsbedingte Verzweigung (2)

GATE x A, B

Zustandsbedingung x:

- FU Facility used
- FNU Facility not used
- SE Storage empty
- SNE Storage not empty
- SF Storage full
- SNF Storage not full
- SA Storage available
- SNA Storage not available
- LR Logic reset
- LS Logic set

*Gate-Block
verwaltet implizit Kette blockierter
Transaktionen,
sortiert nach Priorität und FIFO bei
Gleichheit*

Operanden:

- Operand **A** Nummer/Name des zu testenden Elementes
- Operand **B** ist Zielblock, falls Zustandsbedingung nicht erfüllt ist, Übergang zum Folgeblock, falls Bedingung erfüllt ist.
(**B** leer, dann Blockierung)

Zustandsbedingte Verzweigung (3)

TEST x A, B, C

Arithmetische Zustandsbedingung x:

- E equal
- NE not equal
- L lower
- LE lower equal
- G greater
- GE greater equal

Operanden:

- Operand **A** erster Vergleichswert
- Operand **B** zweiter Vergleichswert
- Operand **C** ist Zielblock, falls Zustandsbedingung nicht erfüllt ist, Übergang zum Folgeblock, falls Bedingung erfüllt ist.
(**C** leer, dann Blockierung)

TEST G C1, 70000

aktuelle Modellzeit

evtl. blockierte Transaktion wird in
RetryChain der Variablen gespeichert

TEST G Q\$Teller, Q\$Teller2, Cont

Warteschlangenlänge

Zyklen

LOOP a, b, c, d

- Operand **a** Parameternummer (Attribut wird zur Zählung der Iterationen benutzt, aktueller Wert wird als Initialwert interpretiert)
- Operand **b** ist der maximale Iteratorwert
- Operand **c** ist die Schrittweite ($a := a + c$)
- Operand **d** ist Zielblock, falls $a \leq b$
sonst Übergang zum Folgeblock, falls maximale Iteration überschritten.

SPEICHER (~ ODEMX-Ressource)

ENTER a

ist ein Block, der die Belegung eines Speichers durch eine Transaktion bewirkt (falls dieser noch frei ist, sonst Blockierung)

- Operand **a** ist Name/Nummer des Speichers, der belegt wird

LEAVE a

ist ein Block, der die Entfernung einer Transaktion aus einem Speicher bewirkt u. die nächste Belegungsblockierung einer Transaktion aufhebt

- Operand **a** ist Name/Nummer des Speichers, der freizugeben ist

Logischer Schalter (Semaphore-Variablen)

LOGIC x A

Belegung x:

- R logic reset (Schalter aus)
- S logic set (Schalter ein)
- I (Schalter umschalten)

Operanden:

- Operand A ist Name/Nummer des Schalters, der belegt wird

*LOGIC-Block
verwaltet implizit Kette blockierter
Transaktionen,
sortiert nach Priorität und FIFO bei
Gleichheit
(Verzögerungskette/DelayChain)*

Einrichtungssperrung

FUNAVAIL A, B, C, D, E, F, G, H

ist ein Block, der die Belegung einer Einrichtung durch Transaktionen verhindert

- Operand **A** ist Name/Nummer der Einrichtung, die gesperrt wird
- Operand **B** ist ein **Modus** {RE- Remove, CO- Continue, leer} der verdrängenden Transaktion
- Operand **C** ist Name/Nummer des Blockes (leer möglich), zu dem die verdrängende Transaktion verzweigt wird
- Operand **D** ist Nummer des Parameters der verdrängten Transaktion, zur Speicherung der Restbedienzeit (kann leer sein)
- Operand **E** ist ein **Modus** {RE- Remove, CO- Continue, leer} der verdrängten Transaktion
- Operand **F** ist Name/Nummer des Blockes (leer möglich), zu dem die verdrängte Transaktion verzweigt wird
- Operand **G** ist ein **Modus** {RE- Remove, CO- Continue, leer} der hängenden oder verzögerten Transaktionen
- Operand **F** ist Name/Nummer des Blockes (leer möglich), zu dem die hängenden oder verzögerten Transaktionen verzweigt werden

Einrichtungsfreigabe

FAVAIL A

- Operand **A** ist Name/Nummer der Einrichtung, deren Sperrung aufgehoben wird.

Speichersperrung und -freigabe

SUNAVAIL A

- Operand **A** ist Name/Nummer des Speichers, der gesperrt wird

SAVAIL A

- Operand **A** ist Name/Nummer des Speichers, dessen Sperrung aufgehoben wird.

Festlegung von Blocknummern (Steueranweisung)

...		
QUEUE	schalterQ	10000
QUEUE	Abteilung	10001
SEIZE	Angestellter1	10002
SEIZE	Angestellter2	10003
ENTER	Bereich	10004
...		

*Blöcke
bekommen interne Nummern*

manchmal für Indizierung ungeeignet

Label EQU A

- Label ist der Name eines Blockes oder eines Skalars
- Operand **A** ist der festzugeordnete Wert

Angestellter1	EQU	1
Angestellter2	EQU	2
Bereich	EQU	3
...		

Value	EQU	10
Angestellter2	EQU	2
Bereich	EQU	3
...		

5. GPSS

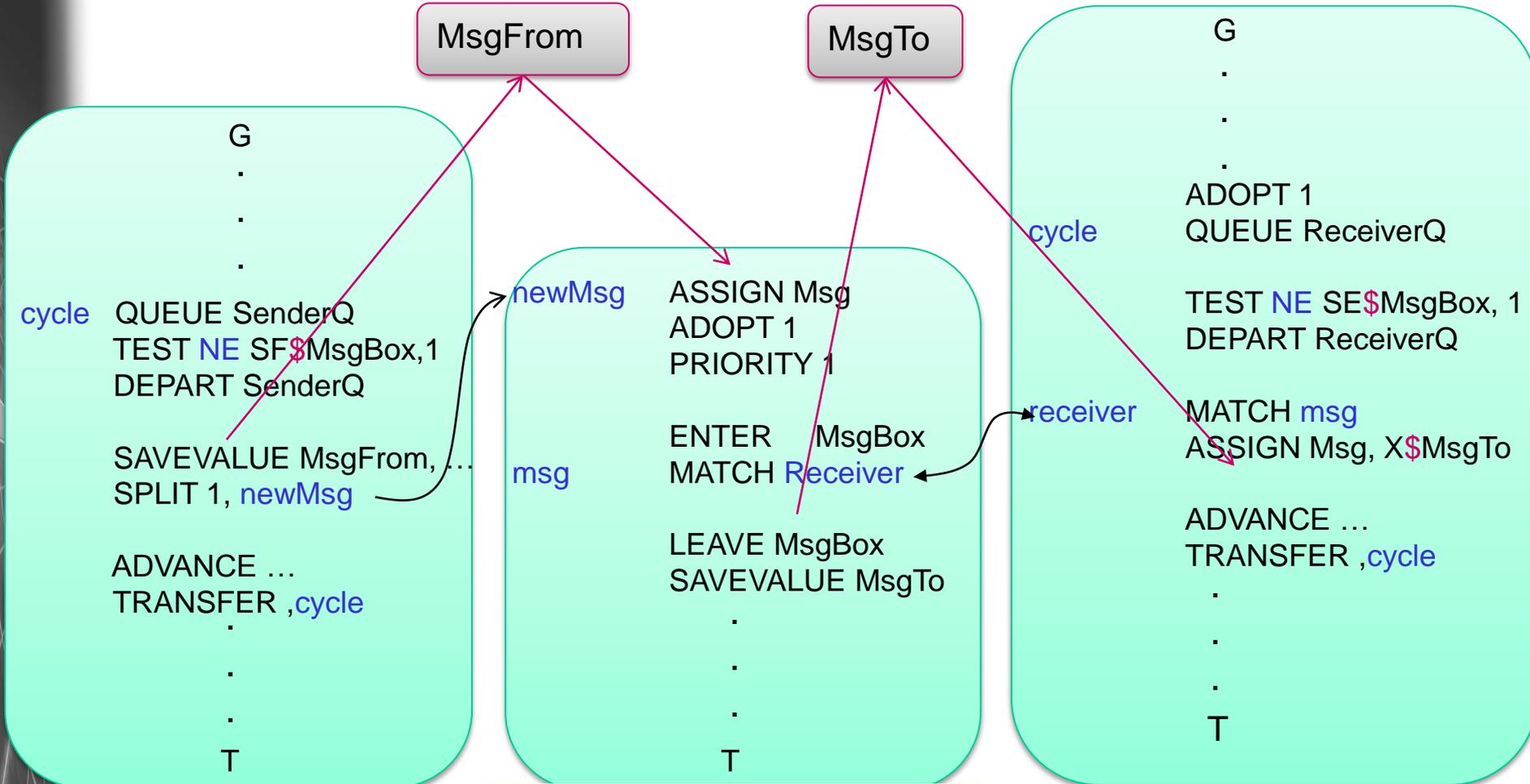
1. Grundphilosophie
2. Block/Stations-Übersicht (1)
3. Einfaches GPSS-Beispiel
4. Erste Einschätzung der GPSS-Ausdruckskraft
5. Vordefinierte Modellierungsmuster (FACILITY)
6. Standardattribute
7. Block/Stations-Übersicht (2)
8. Nutzereigene Modellierungsmuster (asynchrone Kommunikation)

Muster asynchrone Kommunikation

SFi = 1, falls Speicher i voll ist, sonst 0
SEi = 1, falls Speicher i leer ist, sonst 0

MsgBox STORAGE 2

begrenzter Puffer



Sender-Transaktionen

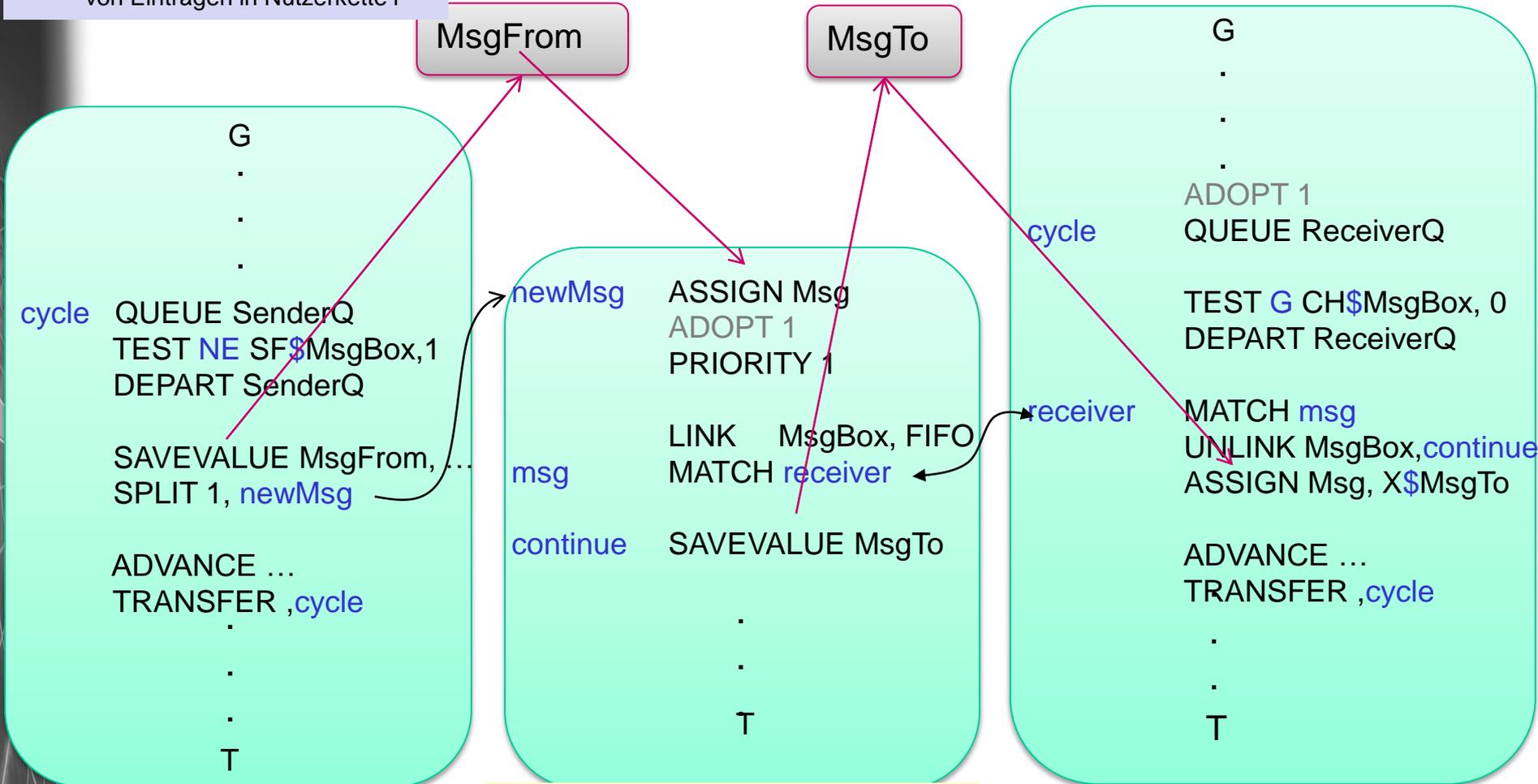
Nachricht-Transaktionen

Empfänger-Transaktionen

Muster asynchrone Kommunikation

SFi = 1, falls Speicher i voll ist, sonst 0
 CHi = augenblickliche Anzahl
 von Einträgen in Nutzerkette i

unbegrenzter Puffer (LINK)



Sender-Transaktionen

Nachricht-Transaktionen

Empfänger-Transaktionen

```

MsgBox                STORAGE  2

                        ** sender
LblSendLoop QUEUE    GENERATE  ,,1
                        SenderQ
                        TEST NE   SF$MsgBox,1
                        DEPART    SenderQ
                        SAVEVALUE rmsg+,1
                        SPLIT     1,LblMsg
                        ADVANCE   2
                        TRANSFER  ,LblSendLoop

LblMsg                ** message
                        ASSIGN    content,X$rmsg
                        ADOPT     1
                        ENTER     MsgBox
                        PRIORITY  1
LblWaitForMsg        MATCH     LblWaitForRecv
                        SAVEVALUE msgContent,P$content
                        LEAVE     MsgBox
                        TERMINATE

                        ** receiver
LblRecvLoop QUEUE    GENERATE  ,,1
                        ADOPT     1
                        ReceiverQ
LblWaitForRecv       TEST NE   SE$MsgBox,1
                        DEPART    ReceiverQ
                        MATCH     LblWaitForMsg
                        ASSIGN    content,X$msgContent
                        ADVANCE   8
                        TRANSFER  ,LblRecvLoop

                        ** sim
GENERATE  ,,200,1
TERMINATE 1

```

```

                        ** sender
LblSendLoop QUEUE    GENERATE  ,,1
                        SenderQ
                        TEST NE   SF$MsgBox,1
                        DEPART    SenderQ
                        SAVEVALUE rmsg+,1
                        SPLIT     1,LblMsg
                        ADVANCE   2
                        TRANSFER  ,LblSendLoop

LblMsg                ** message
                        ASSIGN    content,X$rmsg
                        ADOPT     1
                        LINK      MsgBox,FIFO
LblMsgSave PRIORITY  2
LblWaitForMsg        MATCH     LblWaitForRecv
                        SAVEVALUE msgContent,P$content
                        TERMINATE

                        ** receiver
LblRecvLoop QUEUE    GENERATE  ,,1
                        PRIORITY  1
                        ADOPT     1
                        ReceiverQ
                        TEST G    CH$MsgBox,0
                        DEPART    ReceiverQ
                        UNLINK    MsgBox,LblMsgSave
LblWaitForRecv       MATCH     LblWaitForMsg
                        ASSIGN    content,X$msgContent
                        ADVANCE   8
                        TRANSFER  ,LblRecvLoop

                        ** sim
GENERATE  ,,200,1
TERMINATE 1

```

Weiterentwicklungen von GPSS

SLX (C, Couroutine)

Transaction-Lebenslauf als Coroutine
verschachtelte Structs als Nachbildungen von
Transactionstruktur-Vererbung

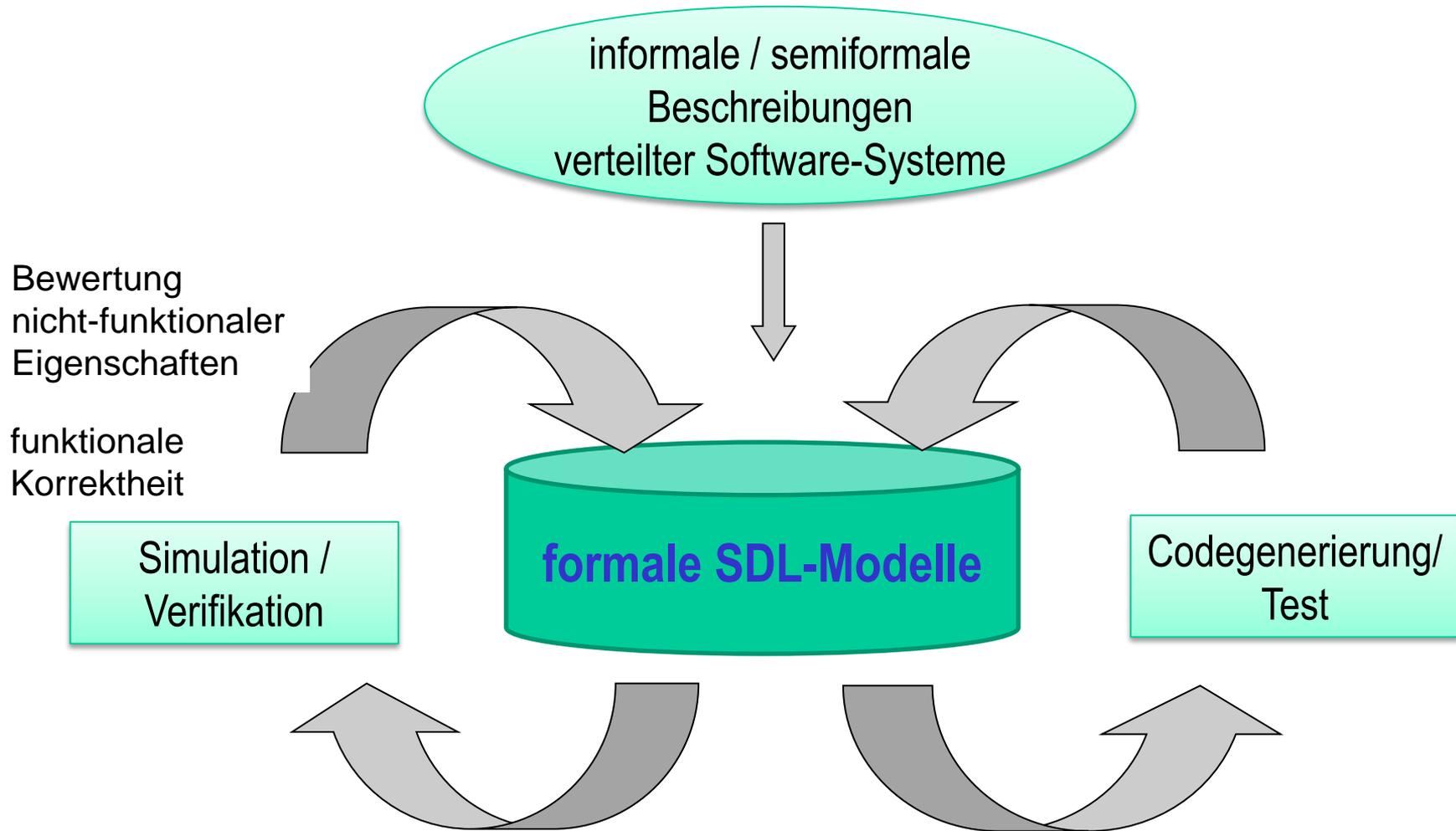
SLX 2.0

Objektorientierte Sicht

6. *SDL*

1. Grundphilosophie
2. ITU-Standard Z.100
3. Werkzeuge
4. SDL-Grundkonzepte
5. Musterbeispiel (in UML-Strukturen)
6. Struktur- und Verhaltensbeschreibung in SDL-RT

SDL= Specification and Description Language



Zur Geschichte

- SDL (Specification and Description Language), Entwicklungsbeginn 1975
standardisiert von der ITU (CCITT), Genf
- verschiedene Versionen (im Abstand von 4 Jahren) Z.100 mit stabilen Kernkonzepten (asynchron kommunizierender Automaten)
- seit 1988 mit Konzepten einer formalen (statischen u. dynamischen Semantik)
- seit 1996 mit objektorientierten Konzepten
- mit SDL-2000 (einheitliches Kalkül für formale Semantik: ASM (Gurjewich, MicroSoft), hierarchische Automaten, OO-DT)
- SDL-2000-Referenz-Compiler (ohne industriellen Nutzen)
- 2005 Wechsel der Kern-Entwickler-Community zur OMG (UML-2.0)
Stabilisierung der UML-Sprache (insbesondere SDL als UML-Profil)

Einstieg der HU Berlin

