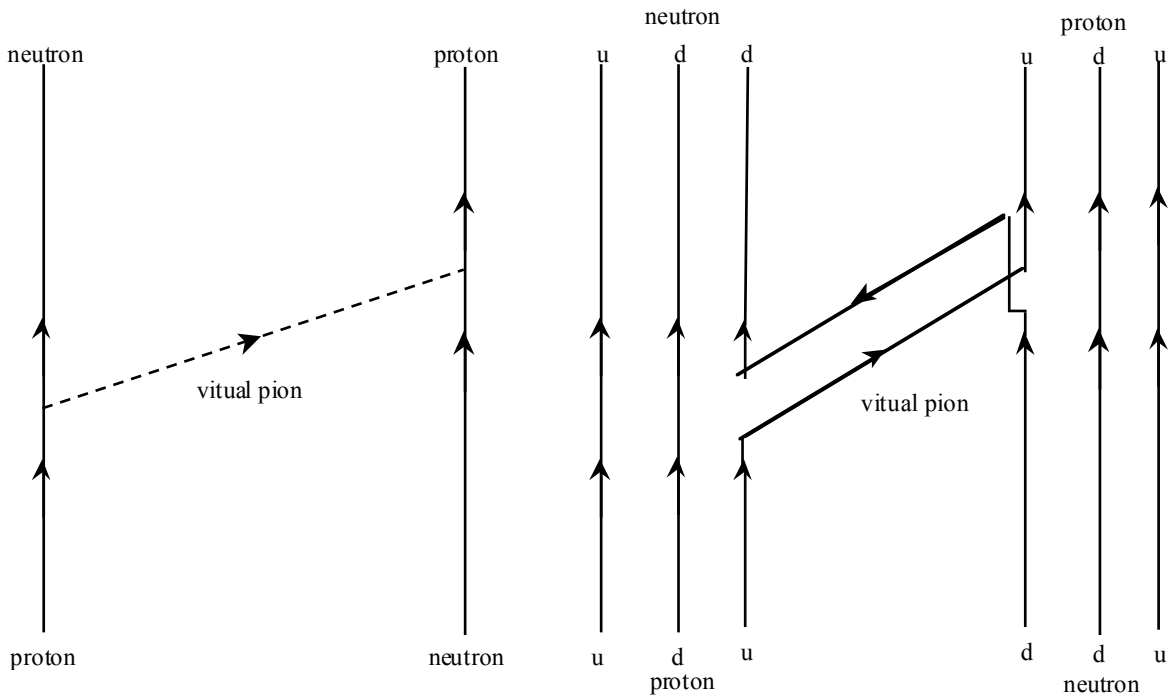
The development of the MSC language - past and future

Ekkart Rudolph

Institut für Informatik, Technische Universität München, D-80290 München, Germany
rudolphe©informatik.tu-muenchen.de

Flow diagrams in physics:

Feynman Diagram with refinement of protons and neutrons into quarks



History

- **SDL-Forum Lisbon, October 1989:**
  First suggestion for MSC standardization

- **CCITT-Meeting in Helsinki, June 1990:**
  MSC standardization decided

- **CCITT-Meeting in Recife, December 1991:**
  Form of Z.120 adjusted to Z.100
  Approval within study group

- **CCITT-Meeting in Geneva, May 1992:**
  Closing session of the study period:
  Approval of recommendation Z.120

- **ITU-TS Meeting in Geneva, November 1993:**
  Revised version of Z.120 (minor corrections)

- **ITU-TS Meeting in Geneva, October 1994:**
  Formal dynamic semantics definition as Annex B to Z.120:
  Message Sequence Charts Algebraic Semantics

- **ITU-TS Meeting in Geneva, September 1995:**
  Formal static semantics definition as Annex C to Z.120:
  Static Semantics of Message Sequence Charts

- **ITU-TS Meeting in Geneva, April 1996:**
  Closing session of the study period:
  Approval of new recommendation Z.120

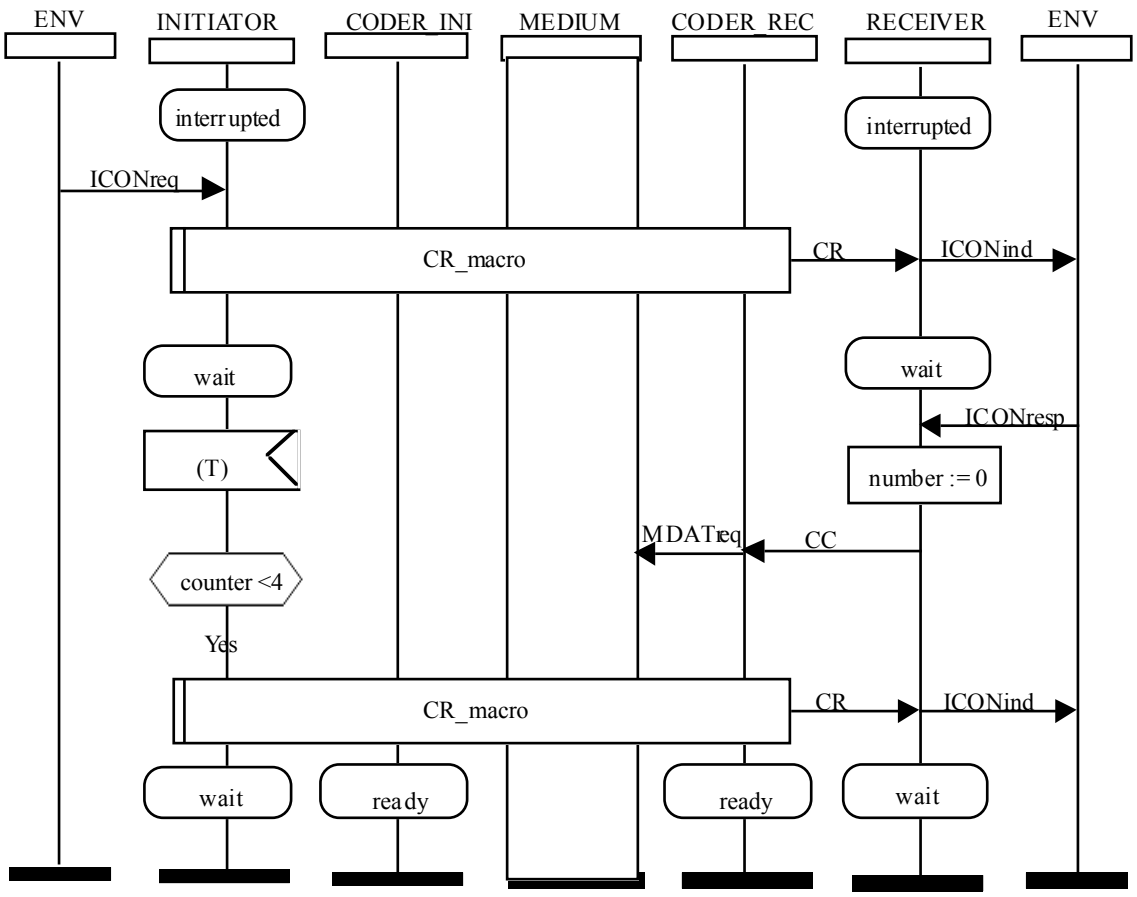
Beginning of MSC standardization: Extended Sequence Charts

SDL - Forum in Lisbon, October 1989

(J. Grabowski, E. Rudolph: Putting Extended Sequence Charts to Practice)

and CCITT experts meeting in Copenhagen, April 1990

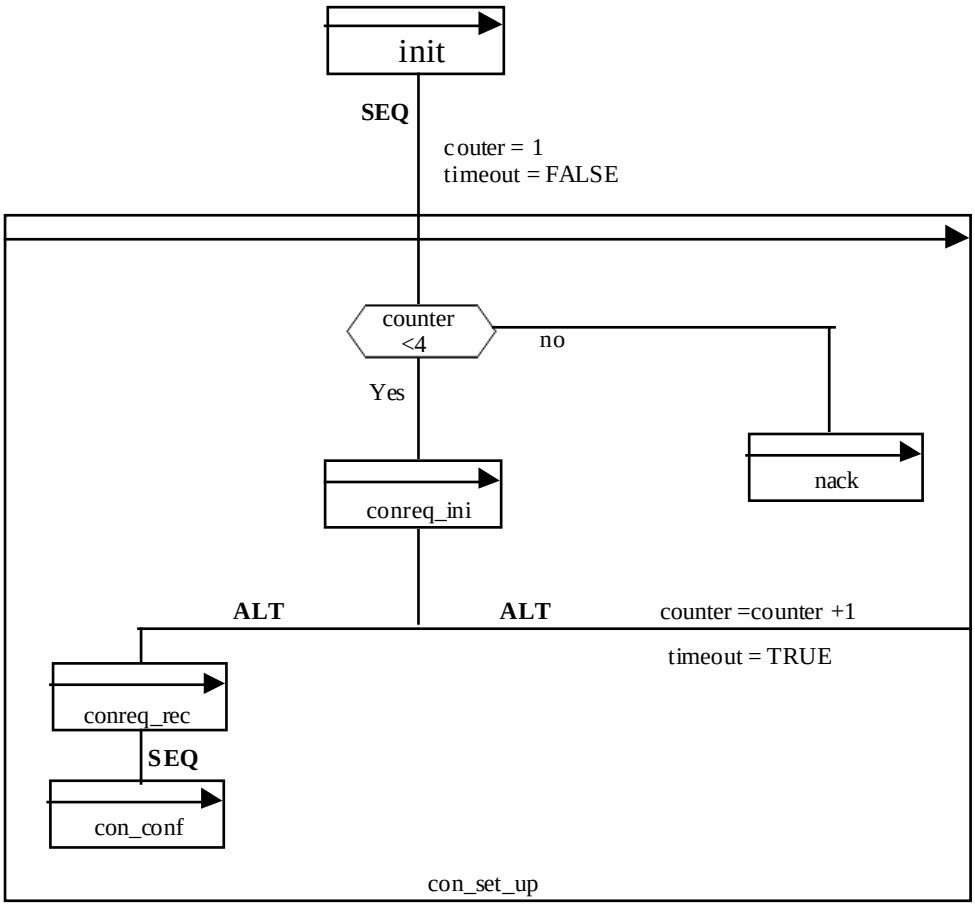
(ESC 90)



Beginning of MSC standardization:

CCITT experts meeting in Copenhagen, April 1990

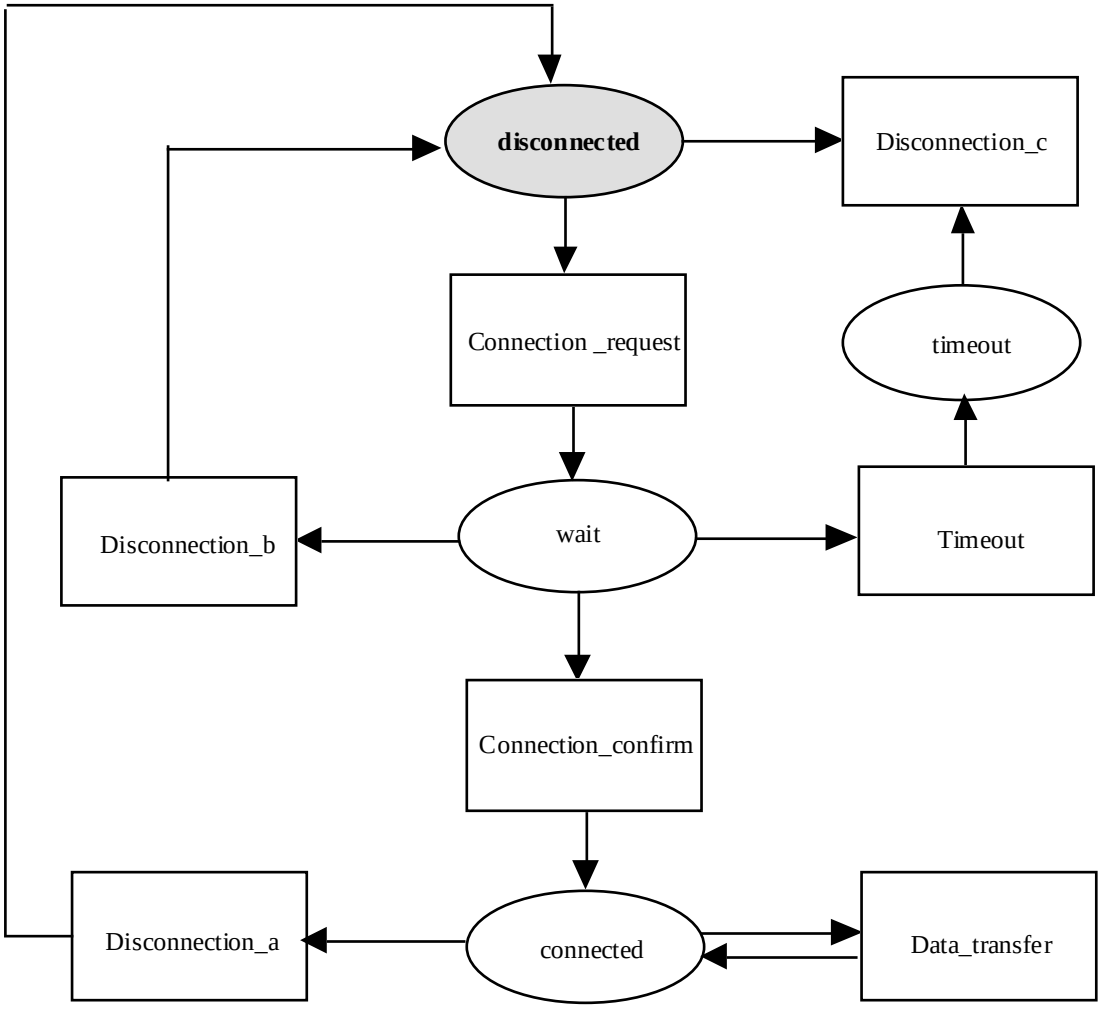
MSC composition using operators



Rudimentary forms of HMSC:

SDL Methodology Guidelines 1992

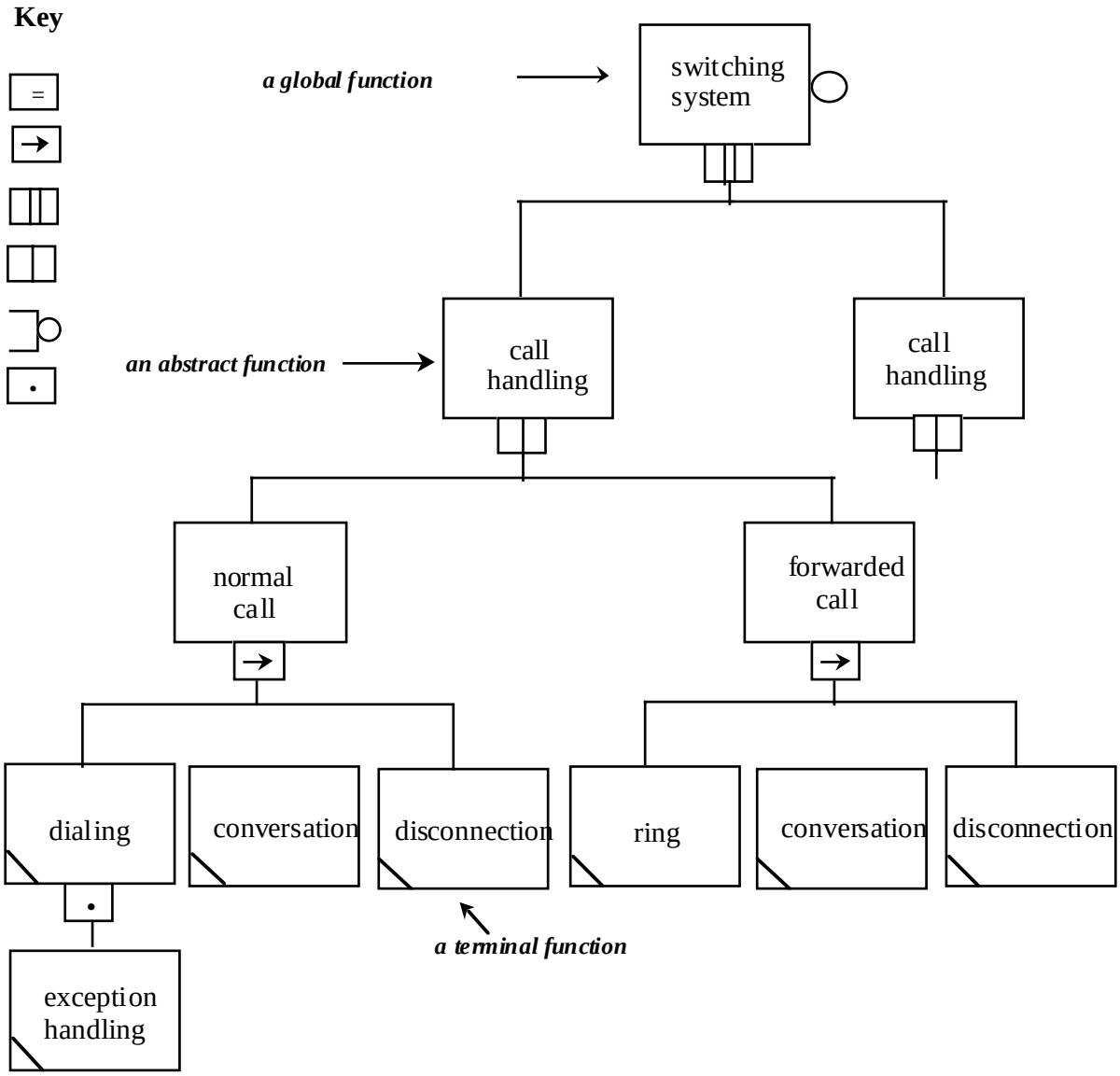
Description of MSC composition by means of an MSC overview diagram



Rudimentary forms of HMSC:

SDL Methodology Guidelines 1992

Functional decomposition - the Verilog variant



Interims Meeting in Geneva, November 1992

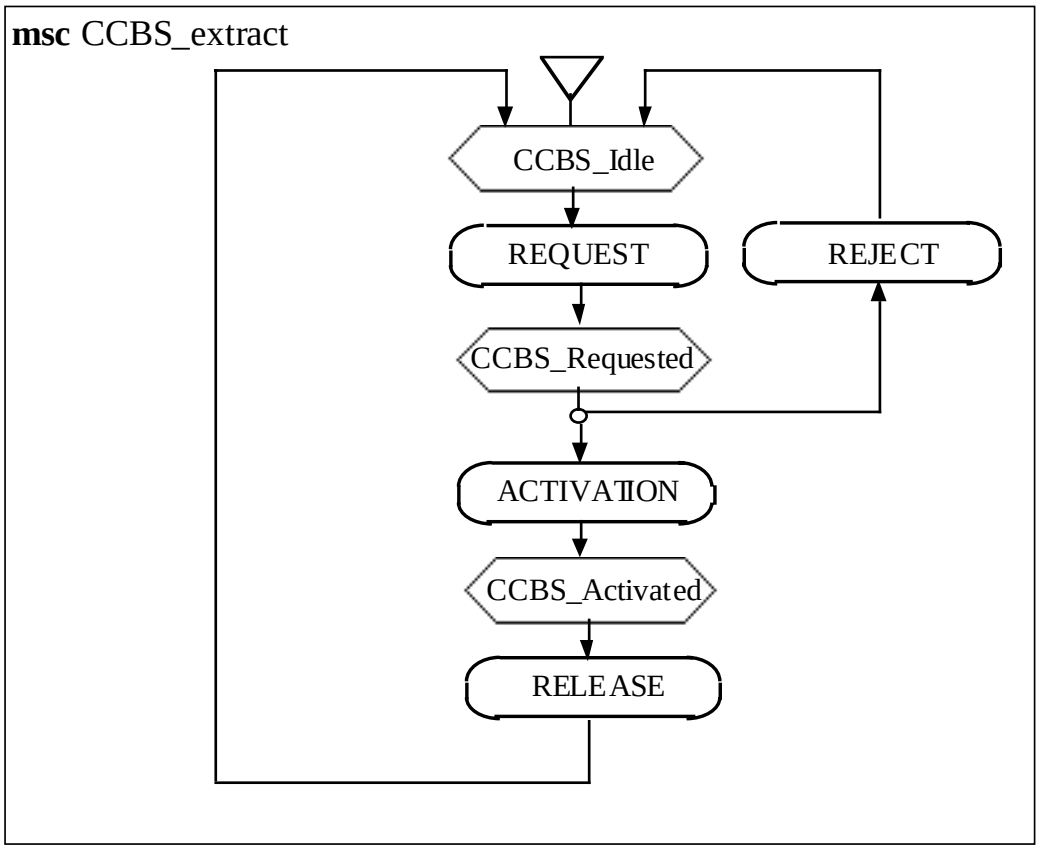
Formal semantics:

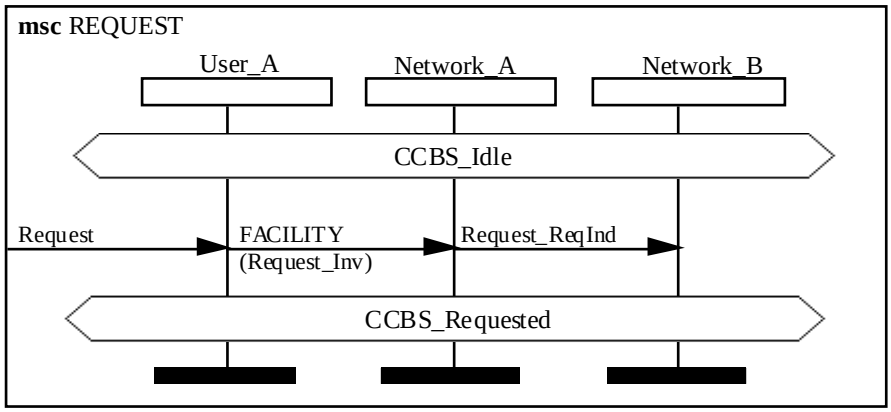
Semantics of synchronous interworkings based on process algebra

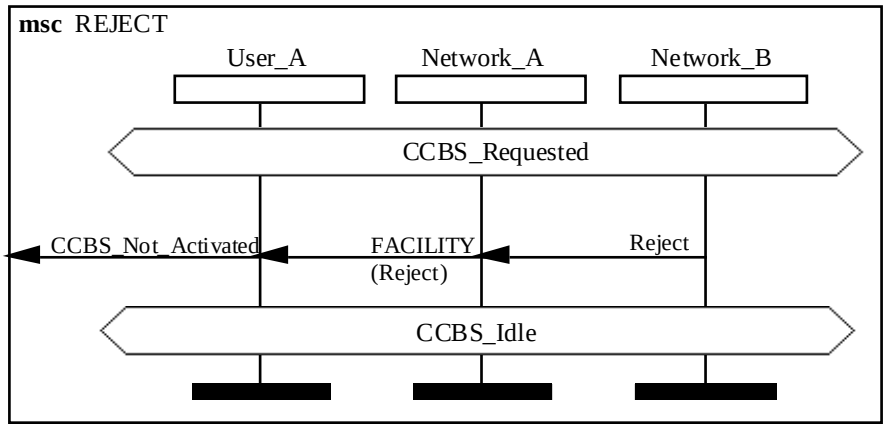
Petri Net based semantics using a special Petri Net class, namely causal or occurrence nets

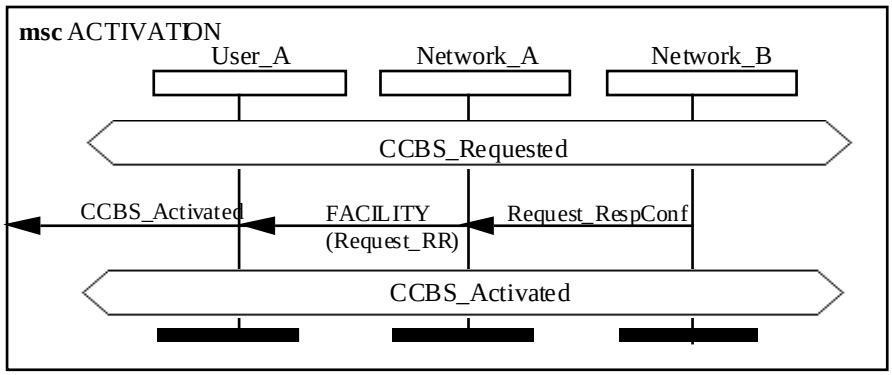
Semantics based on graphs, Buechi automata and temporal logic

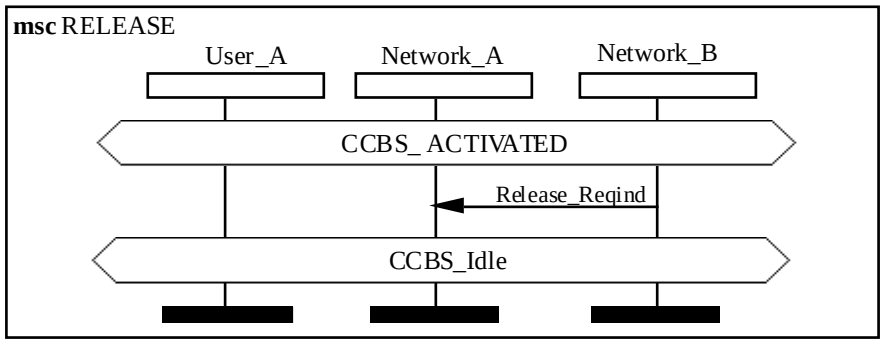
High Level MSC describing the composition of MSCs based on global conditions



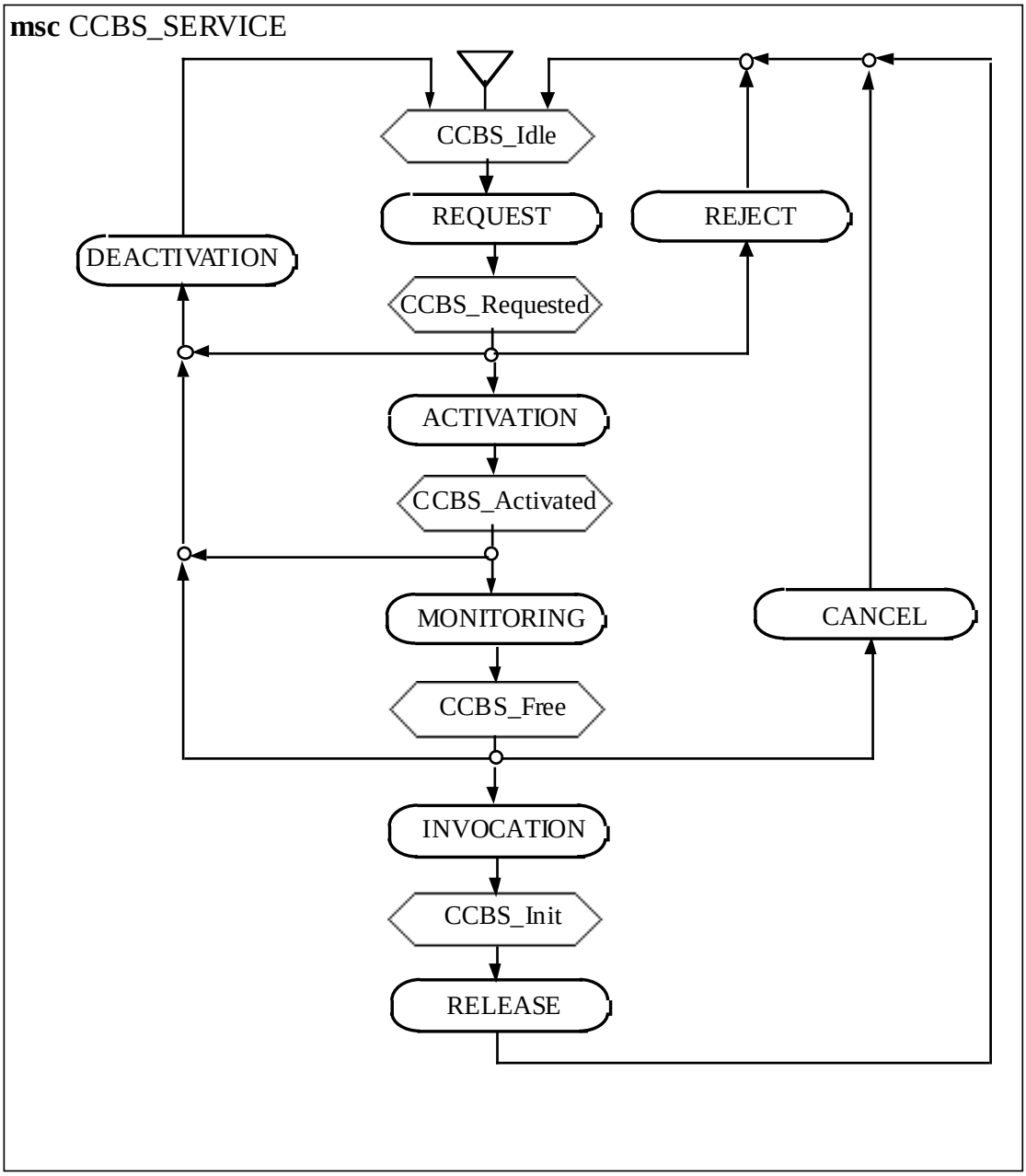

msc REQUEST
User_A
Network_A
Network_B
CCBS_Idle
Request
FACILITY
(Request_Inv)
Request_ReqInd
CCBS_Requested


msc REJECT
User_A
Network_A
Network_B
CCBS_Requested
CCBS_Not_Activated
FACILITY
(Reject)
Reject
CCBS_Idle

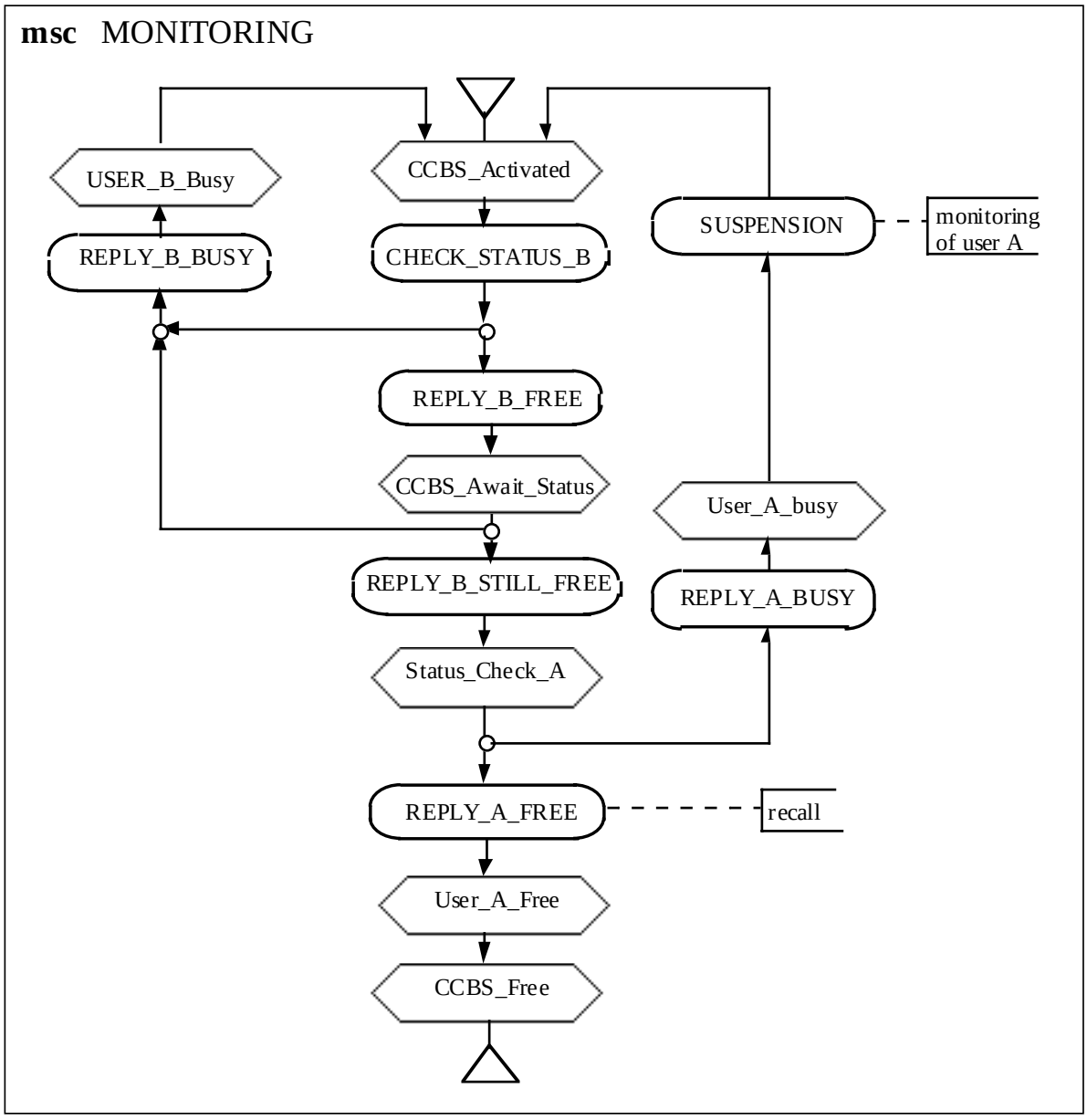

msc ACTIVATION
User_A
Network_A
Network_B
CCBS_Requested
CCBS_Activated
FACILITY
(Request_RR)
Request_RespConf
CCBS_Activated


msc RELEASE
User_A
Network_A
Network_B
CCBS_ ACTIVATED
Release_Reqind
CCBS_Idle

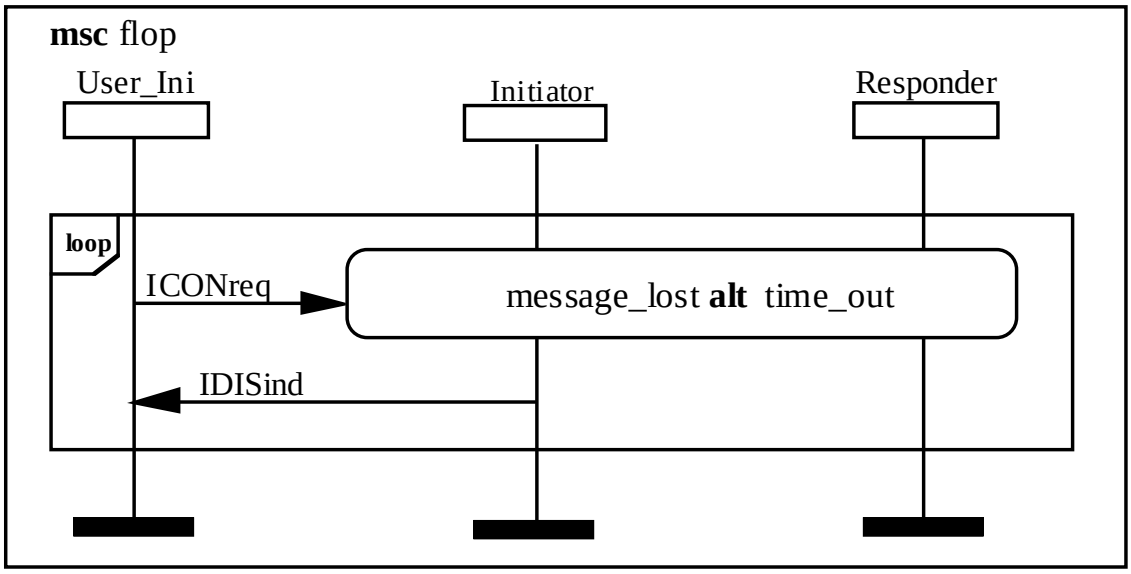
Hierarchical HMSC



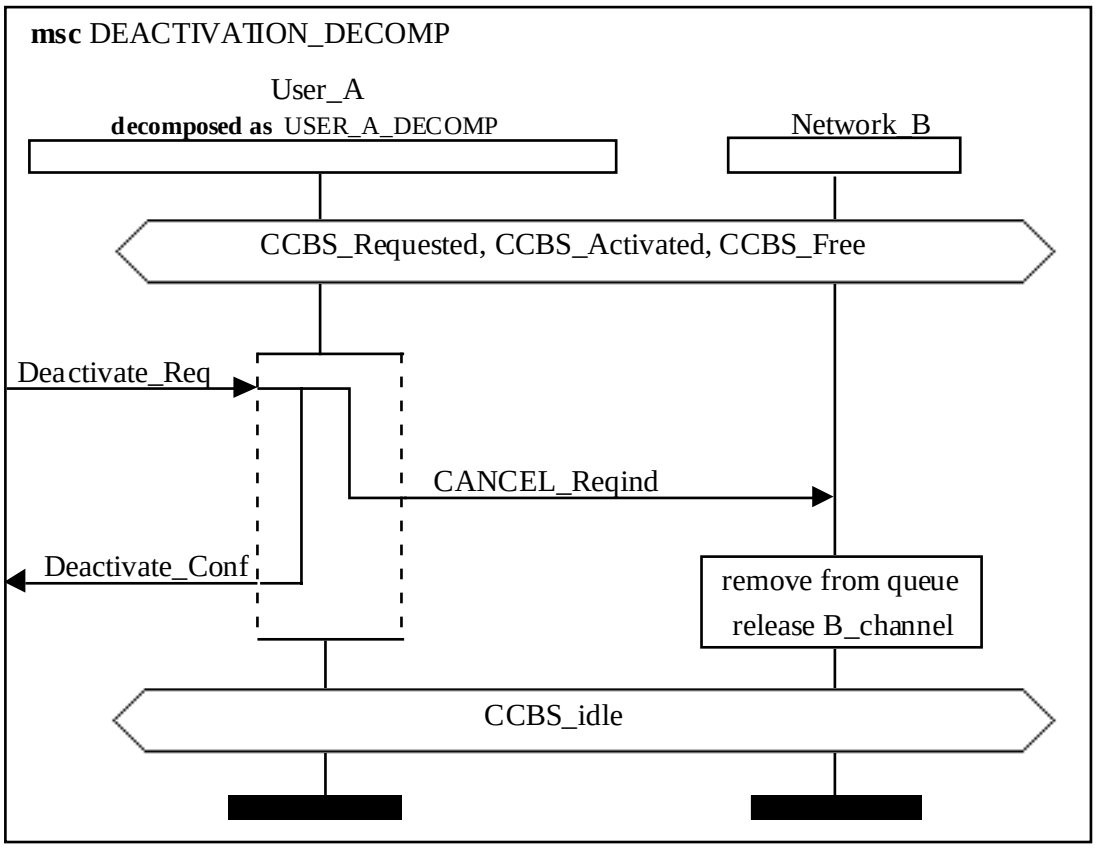
The development of the MSC language - past and future

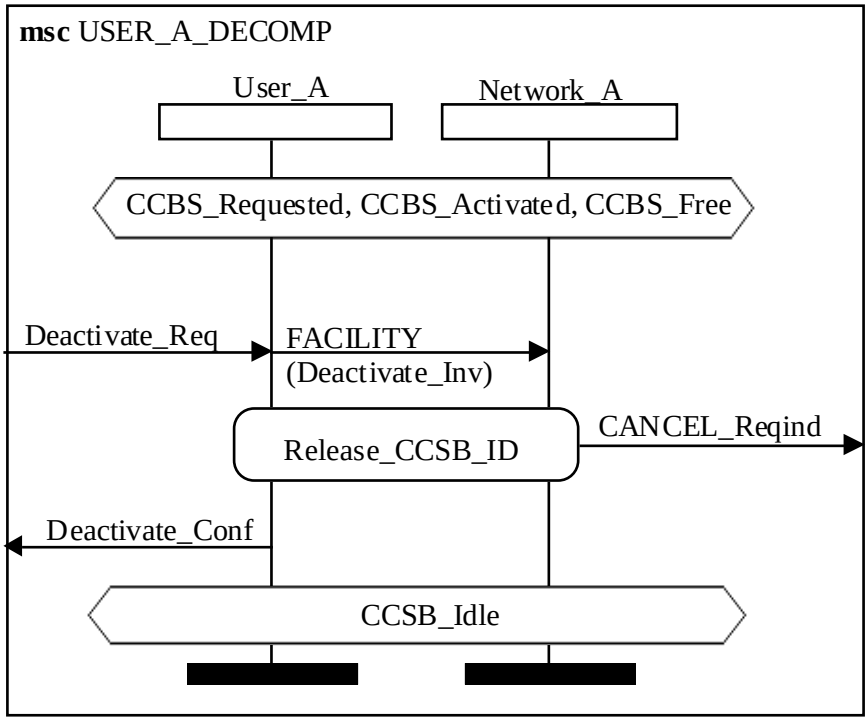

msc MONITORING
USER_B_Busy
CCBS_Activated
SUSPENSION
monitoring of user A
REPLY_B_BUSY
CHECK_STATUS_B
REPLY_B_FREE
CCBS_Await_Status
User_A_busy
REPLY_B_STILL_FREE
REPLY_A_BUSY
Status_Check_A
REPLY_A_FREE
recall
User_A_Free
CCBS_Free

Inline expression and MSC reference expression

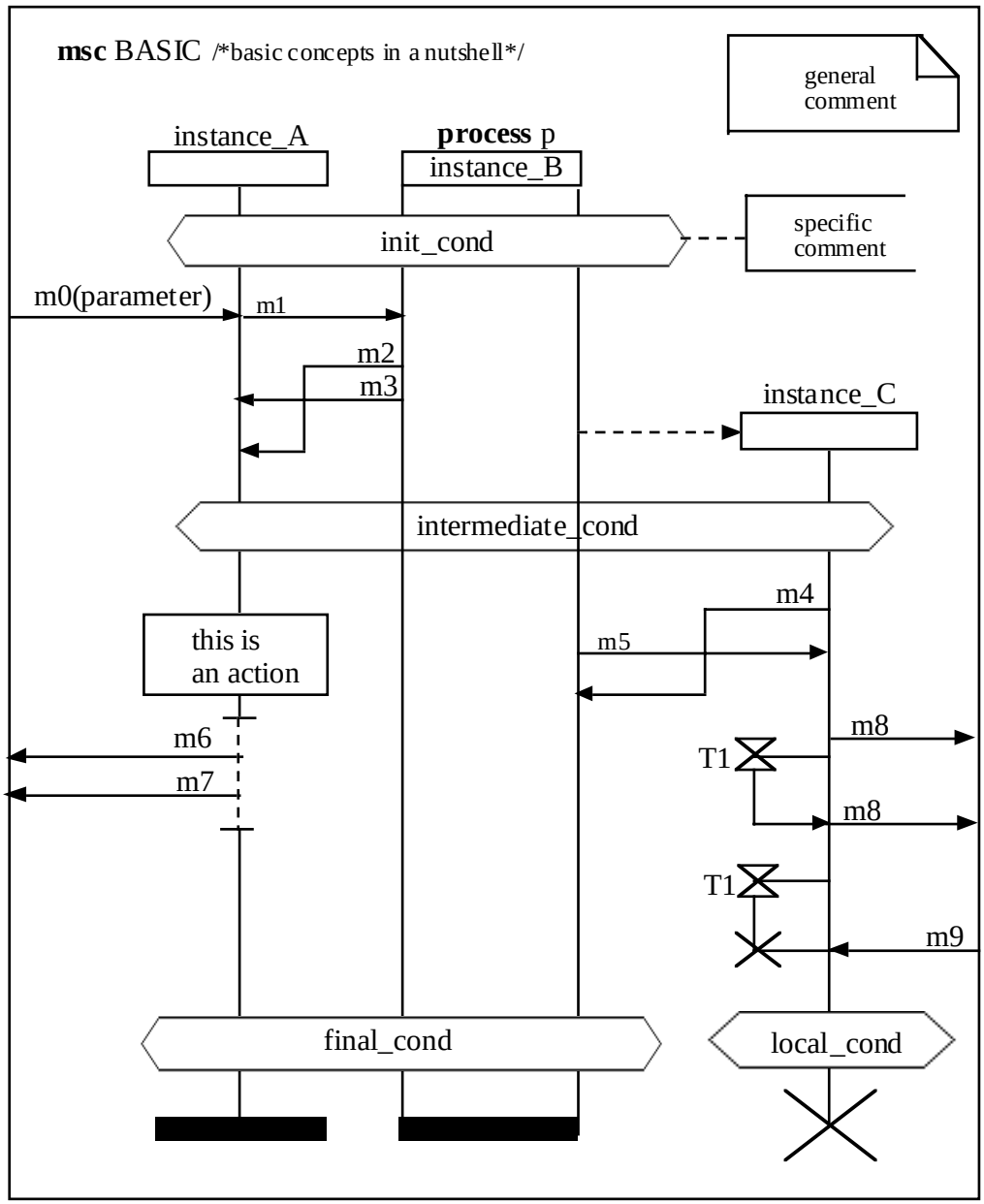


Instance decomposition and generalized ordering

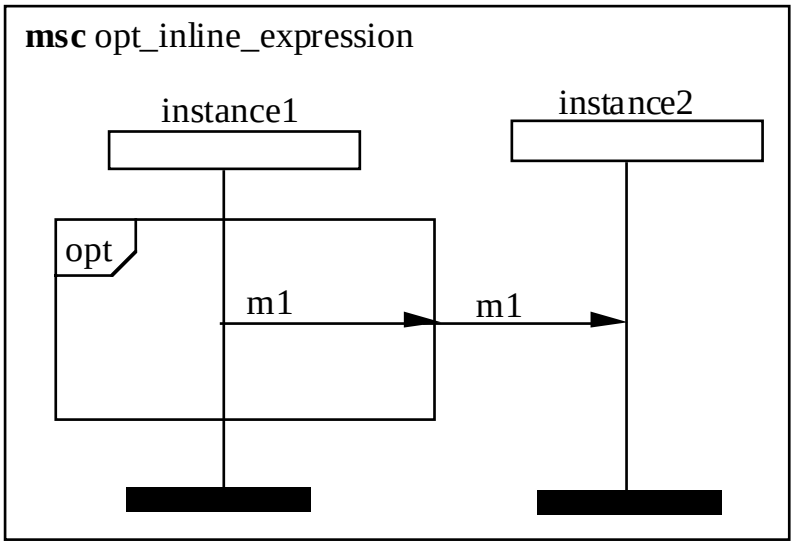


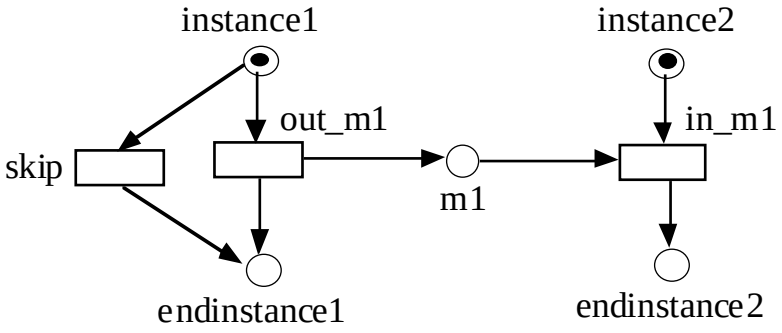


Basic concepts

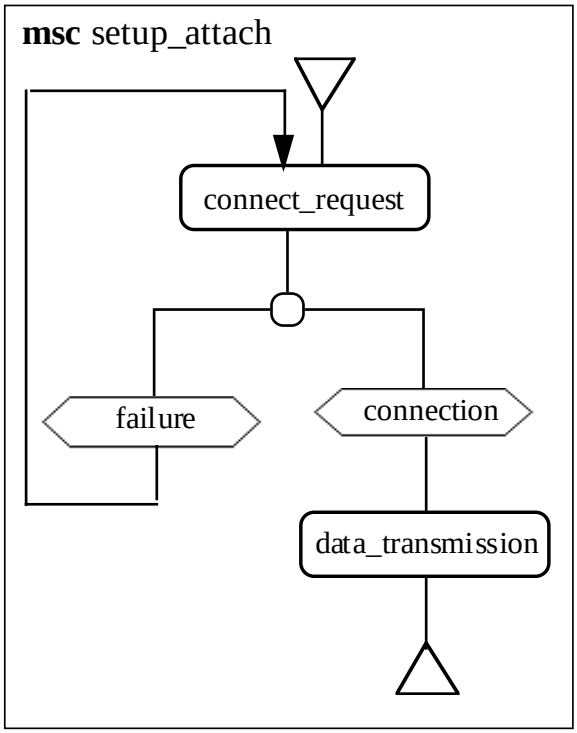


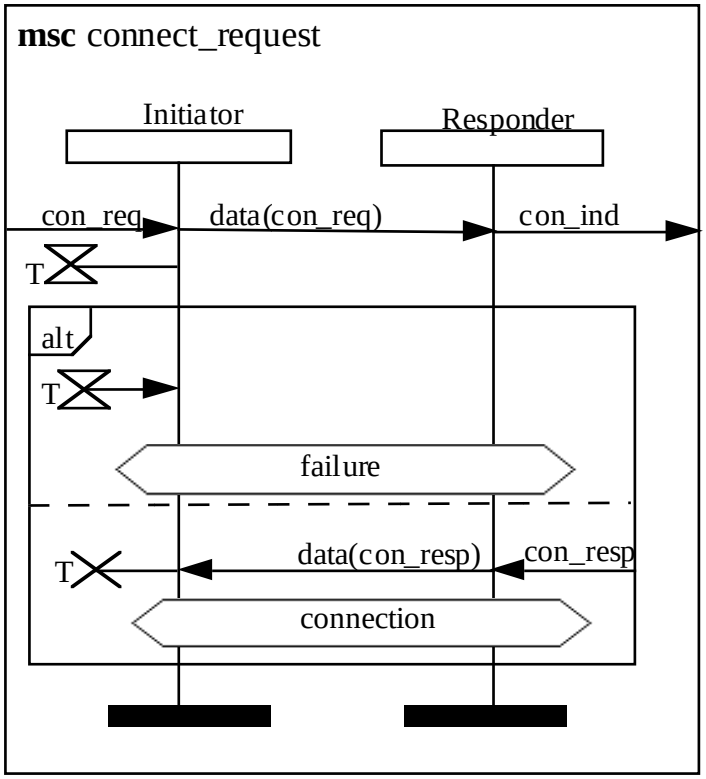
Inline expression containing a deadlock





Conditions as guards





Question 9 - working program (study period 1997 -2000)

Open items to Z.120 to be studied

Below we have listed a number of areas where we know that further study of MSC could improve MSC in the future. The points listed below the area headlines are examples of what subjects we would study under the area, but those subjects are not meant to be excluding other topics in the areas.

1. Non-functional properties

- real-Time descriptions such as duration
- quality of Service properties such as performance, error rates etc

2. Methodology

- use of MSC in object-oriented modelling e.g. formalizing use cases
- test case specifications
- issues related to the use of MSC in close connection with SDL e.g. timer parameters

3. Data concepts

- use of formal data definitions in messages, parameters, conditions and actions

4. Grammars and exchange formats

- improvement of the graphical grammar based e.g. on the study of graph grammar formalisms
- revision of textual grammars including the production of a Common Interchange Format for MSC

5. Conditions

- strong global condition concept
- general predicates in conditions
- further investigation of the relation between composition mechanisms based on conditions and those based on process algebra operators

6. Other language issues

- remote procedure
- synchronous communication construct
- grouping of instances
- hierarchy of messages
- additional MSC operators e.g. disruption, interruption
- total ordering of events
- gates in HMSCs

MSC-2000

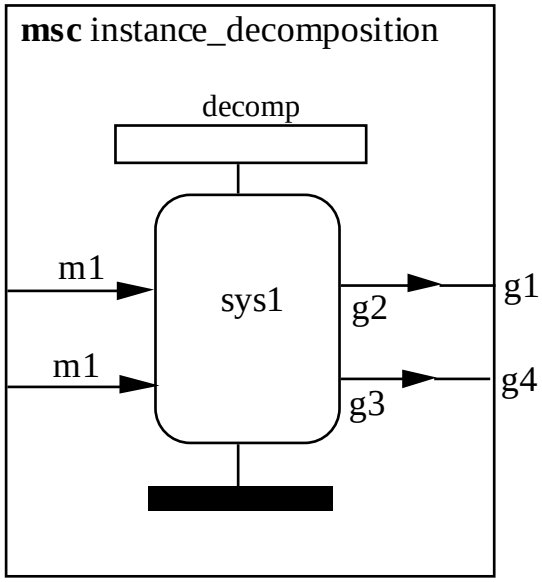
- **Control logics**
  1. **Formal description of data**
  2. **Guards on expression alternatives**
  3. **Break-out of loops**
  4. **Continuation conditions**

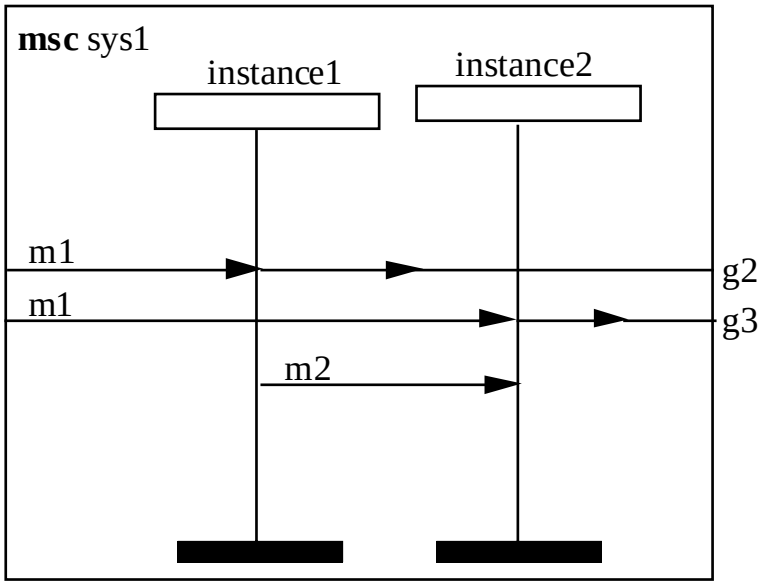
- **Decomposition**
  1. **Inline expressions and decomposition**
  2. **MSC references and decomposition**
  3. **Conditions and decomposition**
  4. **Interface of the decomposition**
  5. **Decomposition of messages**
  6. **Defining a hierarchy of instances**

- **Advanced communication primitives**
  1. **Message multicast**
  2. **Synchronous messages**
  3. **Remote procedures**
  4. **Broadcast**

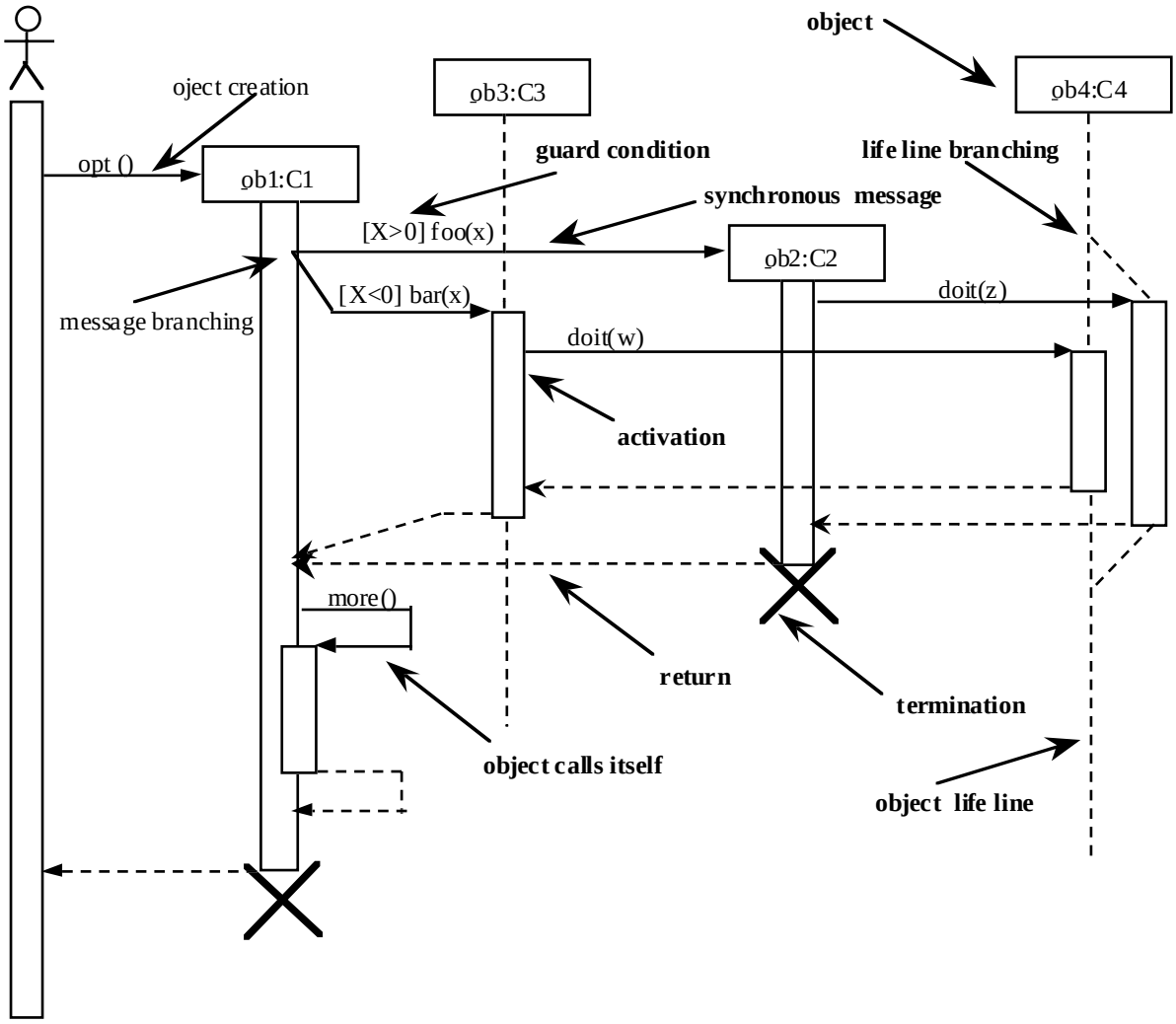
- **Real time constructs**

Instance decomposition described by an MSC reference

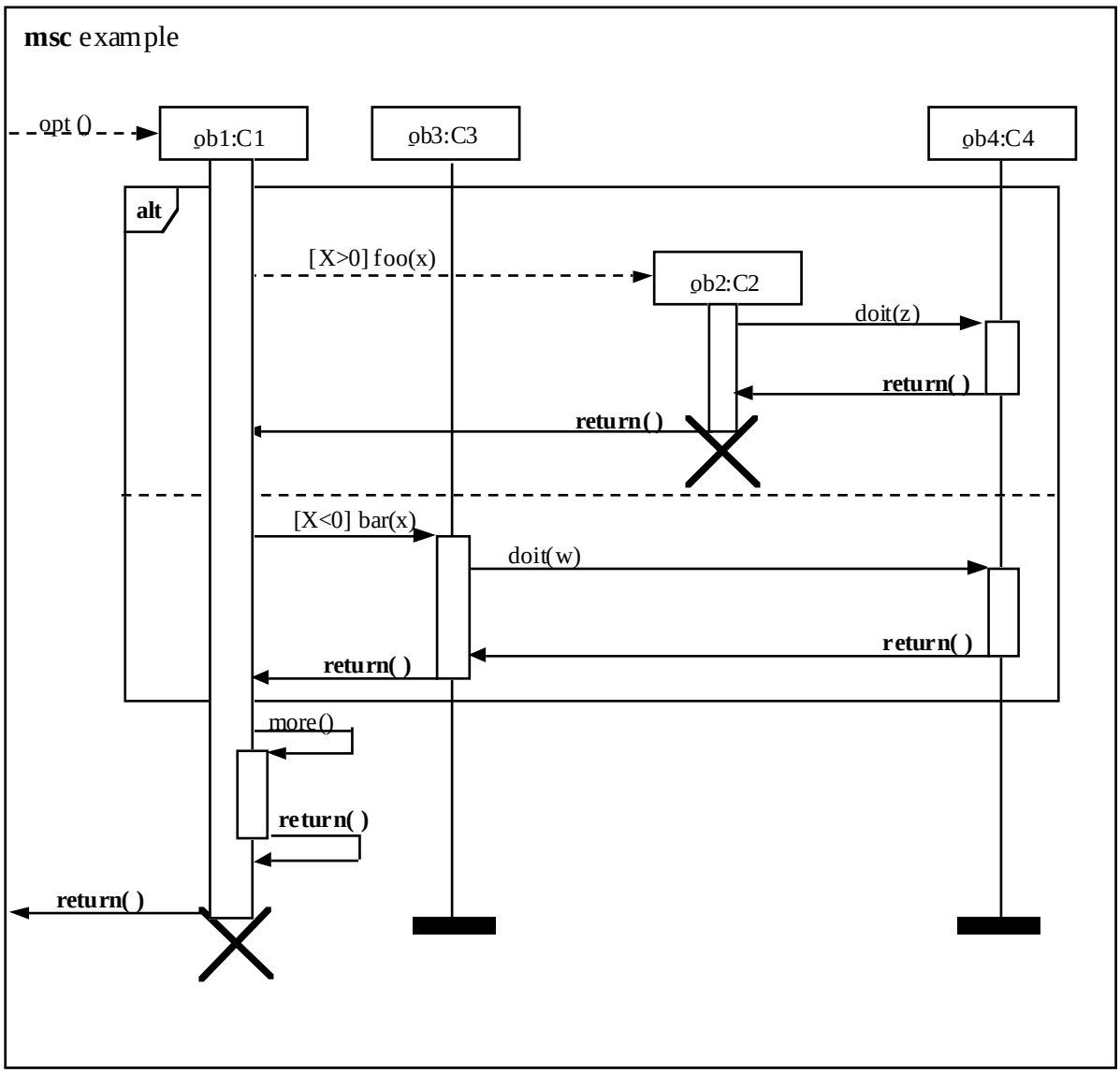




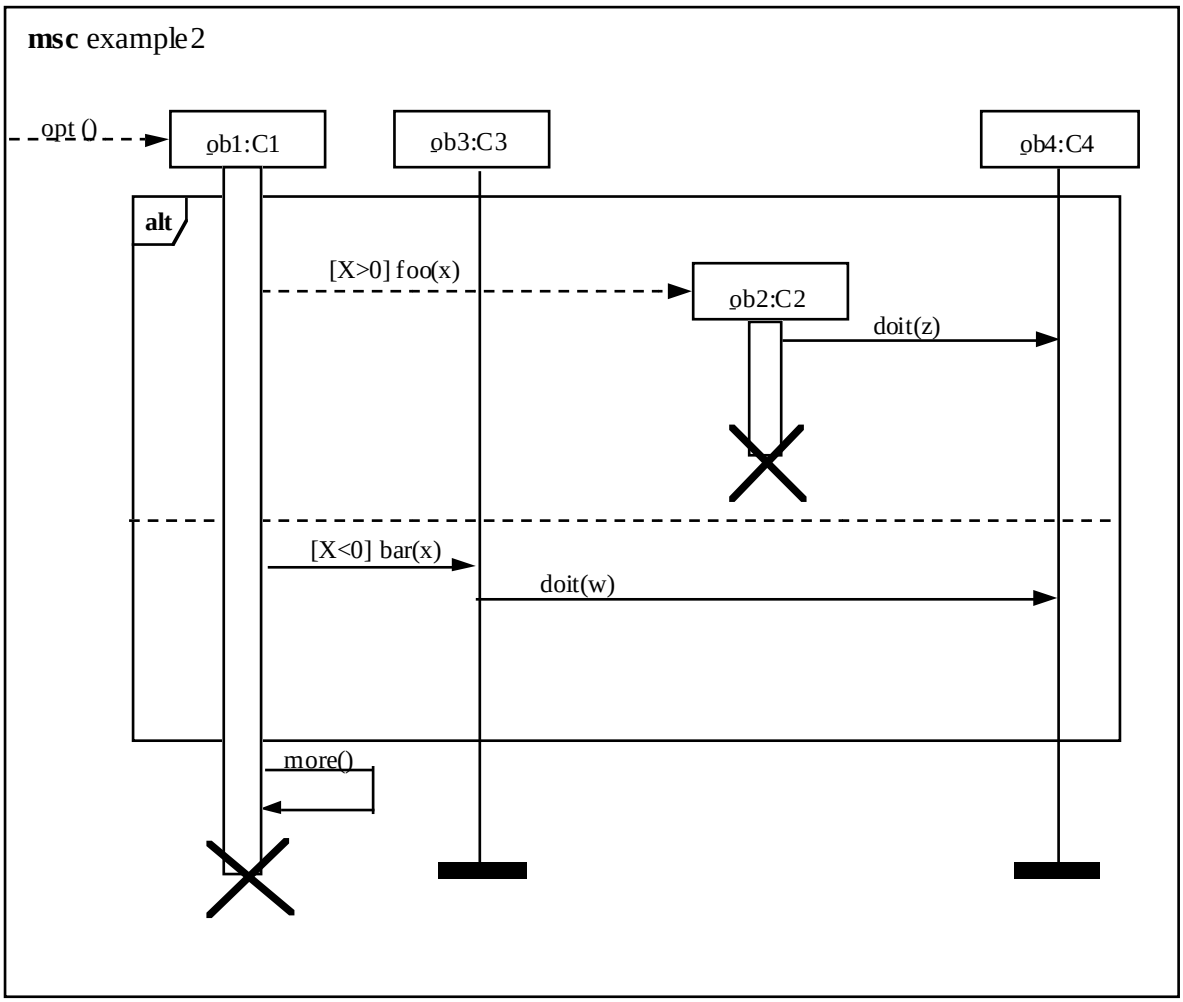
UML - Sequence Diagram



Transformation into extended MSC-96



Transformation into extended MSC-96 without construct for remote procedure call



MSC Quo Vadis?

Message Sequence Charts

Or

Messy Sequence Charts