Tutorial on Message Sequence Charts (MSC'96)

Ekkart Rudolph
Technical University of Munich
Institute for Informatics
Arcisstr.21
D-80290 München
Germany
eMail: rudolphe@informatik.tu-muenchen.de

Jens Grabowski
University of Lübeck
Institute for Telematics
Ratzeburger Allee 160
D-23538 Lübeck
Germany
eMail: jens@itm.mu-luebeck.de

Peter Graubmann
Siemens AG
ZFE TSE
Otto-Hahn-Ring
D-81739 München
Germany
eMail: gr@zfe.siemens.de
History

• 1492:
  Christopher Columbus discovers America

• 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
MSC'96

Anders Ek (Teklogic): Associate Rapporteur for Basic Concepts

Oystein Haugen (Norwegian Computing Center): Associate Rapporteur for Structural Concepts

Sjouke Mauw (Eindhoven University of Technology): Associate Rapporteur for Formal Semantics

Ekkart Rudolph (TUM): Rapporteur

8 - 12 January 1996: Editorial Meeting in Munich (TUM)

10 February 1996: 'White Document' -> ITU


9 - 18 October 1996: World Telecommunication Standardization Conference/ Final Approval

Message Sequence Charts
Message Sequence Charts

```
msc CCBS_SERVICE

DEACTIVATION

REQUEST

ACTIVATION

RELEASE

MONITORING

INVOCATION

CANCEL

Desde CCBS_Service:

CCBS_Idle

CCBS_Requested

CCBS_Activated

CCBS_Free

CCBS_Init
```
**Message Sequence Charts**

```plaintext
msc REQUEST; inst User_A, Network_A, Network_B;
   User_A: instance;
   Network_A: instance;
   Network_B: instance;
   all: condition CCBS_Idle;
   User_A: in Request from env;
      out FACILITY (Request_Inv) to Network_A;
   Network_A: in FACILITY (Request_Inv) from User_A;
      out Request_ReqInd to Network_B;
   Network_B: in Request_ReqInd from Network_A;
      action 'check_queue and service subscription';
      action 'check_compatibility';
   all: condition CCBS_Requested;
   User_A: endinstance;
   Network_A: endinstance;
   Network_B: endinstance;
endmsc;
```
**Message Sequence Charts**

### msc REJECT

- User_A
- Network_A
- Network_B

- CCBS_Request

- CCBS_Not_Activated

- FACILITY (Reject)

- Reject

- CCBS_Idle

### msc ACTIVATION

- User_A
- Network_A
- Network_B

- CCBS_Request

- CCBS_Activated

- FACILITY (Request_RR)

- Request_RespConf

- add to queue

- T-CCBS\(\neq\)

- duration shall be between 15 and 10 minutes

- CCBS_Activated
Monitored monitoring of user A

 MSC MONITORING

USER_B_BUSY
REPLY_B_BUSY

CCBS_Activated
CHECK_STATUS_B

REPLY_B_FREE

CCBS_Await_Status

REPLY_B_STILL_FREE

Status_Check_A

REPLY_A_FREE

User_A_Free

CCBS_Free

SUSPENSION

User_A_BUSY

REPLY_A_BUSY

Recall
Message Sequence Charts
Message Sequence Charts
**Message Sequence Charts**

**FORTE/PSTV’96**

---

** MSC REPLY_A_BUSY **

- **User_A**
- **Network_A**
- **Network_B**

**Status_Check_A**

- **alt**

**FACILITY**

- (StatusRequest_busy)

**T_CCBS1**

**FACILITY**

- (BFree_Inv)

**Suspend**

**suspend request in queue**

**release B_channel reservation**

**User_A_Busy**

---
\textbf{msc \texttt{REPLY\_A\_BUSY\_ALT1}}

\begin{itemize}
  \item \texttt{Facility (StatusRequest\_busy)}
  \item \texttt{Facility (StatusRequest\_busy)}
  \item \texttt{Facility (StatusRequest\_busy)}
  \item \texttt{T\_CCBS1}
\end{itemize}

\begin{itemize}
  \item \texttt{Suspend request in queue}
  \item \texttt{release B\_channel reservation}
\end{itemize}

\textbf{Message Sequence Charts}
Message Sequence Charts

```
msc REPLY_A_BUSY_ALT2

User_A

Network_A

Network_B

Status_Check_A

FACILITY (StatusRequest_busy) g1

FACILITY (StatusRequest_busy)

alt

T_CCBS1

FACILITY (BFree_Inv)

Suspend

suspend request in queue

release B_channel reservation

User_A_Busy
```
Message Sequence Charts
Message Sequence Charts

**msc INVOCATION**

User_A  Network_A  Network_B

Recall  SETUP(Call_Inv)  \(T_{CCBS3}\)

Call(Call_Inv)

CCBS_Free

**msc RELEASE**

User_A  Network_A  Network_B

CCBS_Init

CANCEL_Reqind

Release_CCBS_ID

remove request from queue

release B-channel reservation

CCBS_Idle
**Message Sequence Charts**
**Message Sequence Charts**

**MSC DEACTIVATION**

- User_A
- Network_A
- Network_B

CCBSRequested, CCBSActivated, CCBSFree

Deactivate Req → FACILITY (Deactivate Inv) → Release CCSB_ID → CANCEL ReqInd → remove from queue release B_channel

Deactivate Conf → CCBS_idle

**MSC RELEASE_CCBS_ID**

- User_A
- Network_A

CCBS Deactivation Requested

FACILITY (Deactivate RR)

opt

- option valid if deactivation is caused by user A

CANCEL ReqInd

- TCCBS2
- TCCBS3

release CCBS reference

release CCBS reference
Message Sequence Charts

**msc BUSY_AGAIN**

- Network_B
- User_B_Free
- T_CSSB4
  - status_request
  - status_busy
- User_B
  - off_hook
  - digit
  - seizure_int
    - ack
    - answer
    - connection

**msc ABSTRACTION**

- CCBS_user
- CCBS_destination
- CCBS_Free
- CCBS_recall
- CCBS_call
- CCBS_Init
Message Sequence Charts

**Deactivation Decomposition (DEACTIVATION_DECOMP):**
- User_A initiates a deactivation request (Deactivate_Req).
- Network_B receives the request and recovers resources (CCBS requested, CCBS activated, CCBS free).
- After processing, Network_B sends a deactivation confirmation (Deactivate_Conf) to remove the request from queue and release B channel.

**User A Decomposition (USER_A_DECOMP):**
- User_A requests deactivation (Deactivate_Req).
- Facility (FACILITY) processes the request (Deactivate_Inv) and sends a release command (Release_CCSB_ID) to Network_A.
- CANCEL_Reqind is sent to acknowledge the cancellation.
- Deactivation completes, and the channel becomes idle (CCSB_Idle).
 MSC CHECK_STATUS

CHECK_STATUS_B

REPLY_B_FREE

CCBS_Await_Status

REPLY_B_BUSY

USER_B_busy

REPLY_B_STILL_FREE

Status_Check_A

REPLY_A_FREE

User_A_Free

REPLY_A_BUSY

User_A_busy

Message Sequence Charts
Message Sequence Charts
**Message Sequence Charts**
message \texttt{REPLY\_B\_STILL\_FREE2}

\begin{tikzpicture}
  \node [rectangle, draw] (User_A) at (0,0) {User\_A};
  \node [rectangle, draw] (Network_A) at (2,0) {Network\_A};
  \node [rectangle, draw] (Network_B) at (4,0) {Network\_B};

  \draw [->] (User_A) -- (Network_A) node [midway, above] {Remote\_User\_Free};
  \draw [->] (Network_A) -- (Network_B) node [midway, above] {STATUS (confirm\_free)};

  \draw [->] (Network_B) -- (User_B_Free) node [midway, above] {User\_B\_Free};
  \draw [->] (User_B_Free) -- (Status\_Check\_A);

  \draw [->] (Status\_Check\_A) -- (Network_B);
\end{tikzpicture}
Outlook

Specific task objectives with expected time frames for completion

1. Revised Annexes B and C (formal semantics) corresponding to MSC-96 should be provided by 2H 97

2. An addendum to Z.120 should be provided by 2H 98

3. A corresponding Addendum of formal semantics (Annex B and Annex C) should be provided by 2H 99

4. A revised recommendation Z.120 should be provided by 2H 2000

5. Revised Annexes B and C (formal semantics) corresponding to MSC-2000 should be provided by 2H 2001
Outlook

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