Message Sequence Charts, Live Sequence Charts

SE Systementwurf WS 05/06

Evgeniya Ershova

Gliederung

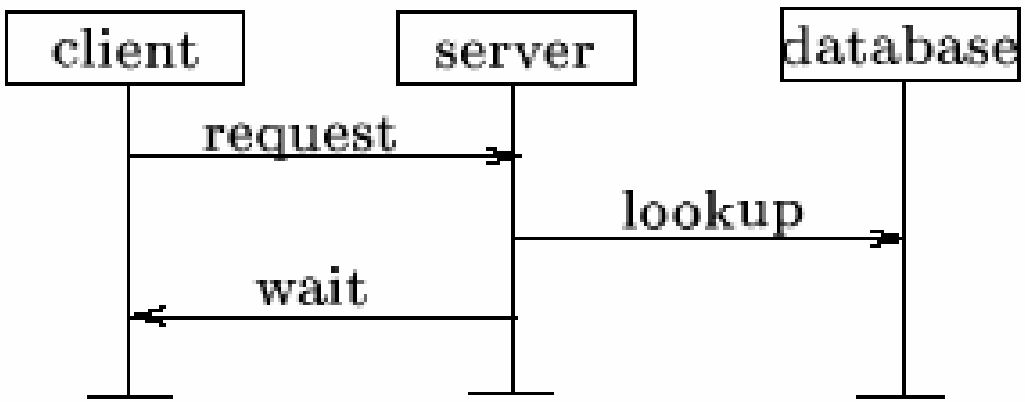
- Heute
  - basic MSC‘s
  - Message Sequence Graphs
  - Hierarchical Message Sequence Charts
  - Anwendung
- 01.12.2005
  - Live Sequence Charts
  - Demo mit Play-Engine

Message Sequence Charts :

- graphische Spezifikationssprache
- dient zur Beschreibung des Kommunikationsverhaltens zwischen Systemkomponenten und deren Umgebung
- ITU-T Standard Z.120
- integriert als *sequence diagramms* in UML

Message Sequence Charts : Definition (1)

- Prozesse (Instanzen) – vertikale Linien
- Nachrichten - horizontale Pfeile, um den Zeitverlauf anzuzeigen



Message Sequence Charts : Definition (2)

- $M = \{P, E, C, l, m, <\}$, *wobei*
  - P - eine Menge von Prozessen
  - E - eine Ereignis-Menge
  - C – eine endliche Namensmenge für Nachrichten und lokale Aktionen
  - $l : E \rightarrow T = \{p!q(a), p?q(a), p(a) \mid p \neq q \in P, a \in C\}$ – Beschriftungsfunktion
    - $p!q(a)$ – p sendet eine Nachricht a nach q
    - $p?q(a)$ – p empfängt eine Nachricht a von q
    - $p(a)$ – ein lokales Ereignis

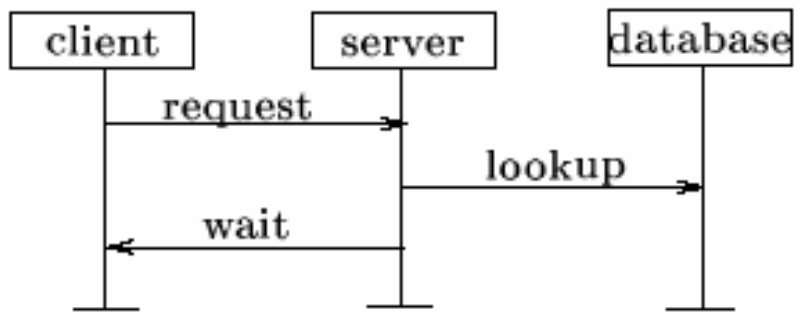
Message Sequence Charts : Definition (3)

- $M = \{P, E, C, l, m, <\}$, *wobei*
  - $m : S \rightarrow R$ - eine bijektive Funktion, die jedem Send-Ereignis das zugehörige Receive- Ereignis zuordnet
  - eine Halbordnung $<_i$ auf den Ereignissen des Prozesses P_i
    - $s < r$, wenn $m(s) = r$
    - legt die zeitliche/visuelle Reihenfolge der Ausführung der Ereignisse fest

Message Sequence Charts : *Linearisierung*

- Die Erweiterung < zu einer totalen Ordnung

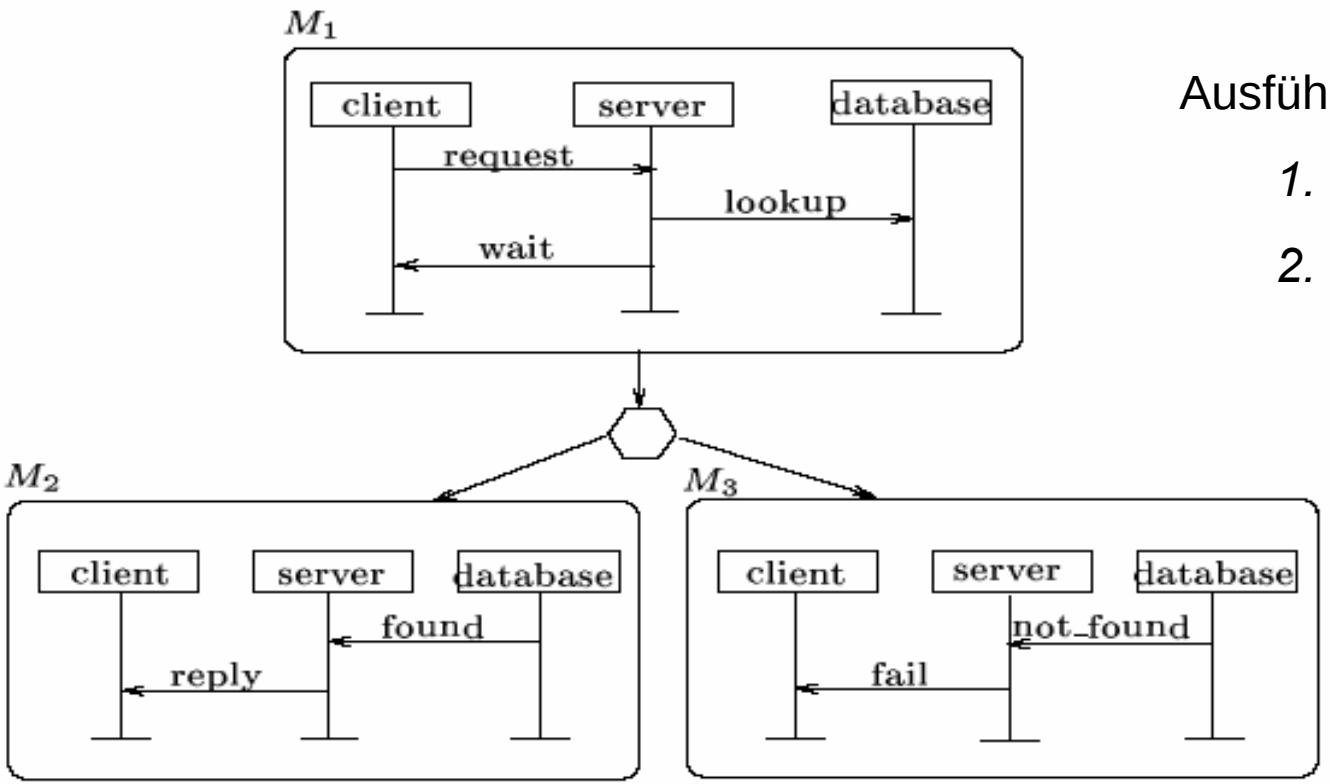
client!server(request)
server?client(request)
server!database(lookup)
database?server(lookup)
server!client(wait)
client?server(wait)



Message Sequence Graph : Definition

- – Graph deren Knoten – MSC‘s und Kanten – Konkatenation von MSC‘s
- $G = \{V, R, v^0, V_f, \lambda\}$, *wobei*
  - V – eine Knotenmenge
  - R – eine Menge der Transitionen $R \subseteq V \times V$
  - $v^0 \in V$ – der Anfangsknoten
  - $V_f \subseteq V$ – die Endknoten
  - λ(v) – Beschriftungsfunktion vom Knoten v

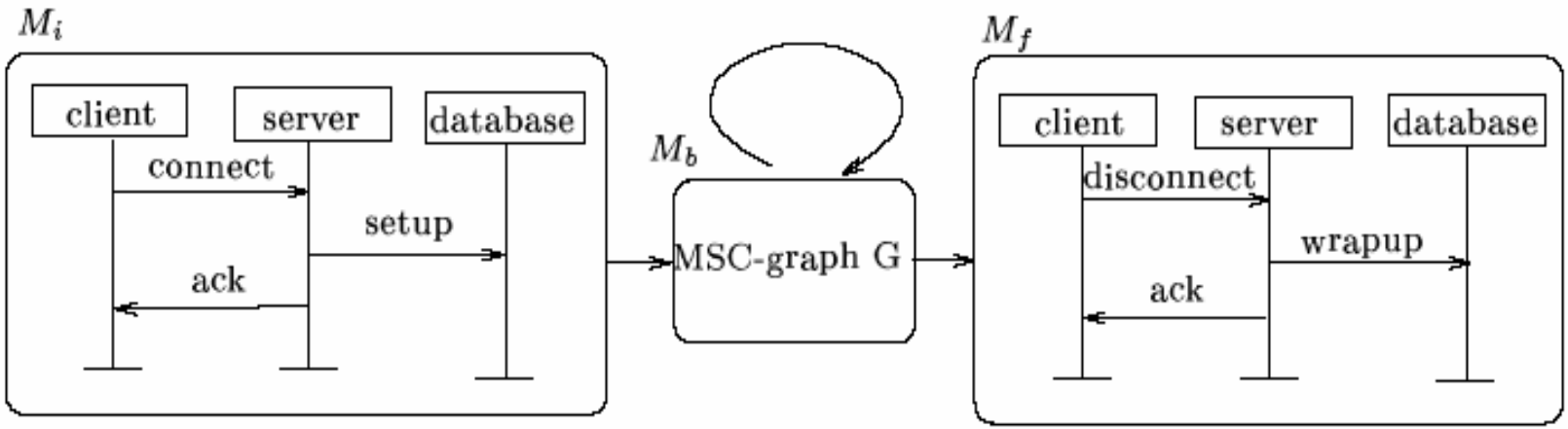
Message Sequence Graph : Beispiel



Ausführung von *G:*

1. M_1M_2
2. M_1M_3

Hierarchical MSC‘s (HMSC)



- *High-level* MSC‘s
- verbesserter Strukturierungsmechanismus
- - Graph deren Knoten – MSC‘s oder andere HMSC‘s $\Rightarrow$ Verschachtelung von Graphen
- Beschrieftung verschiedener Knoten mit gleichen HMSC‘s

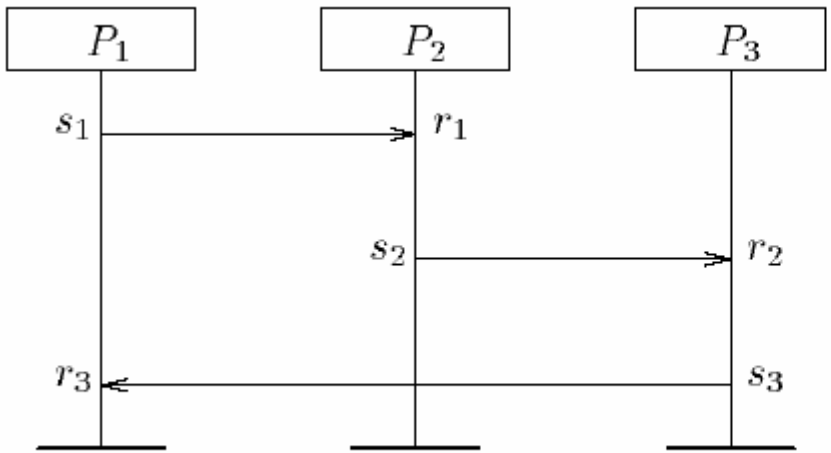
HMSC : Definition

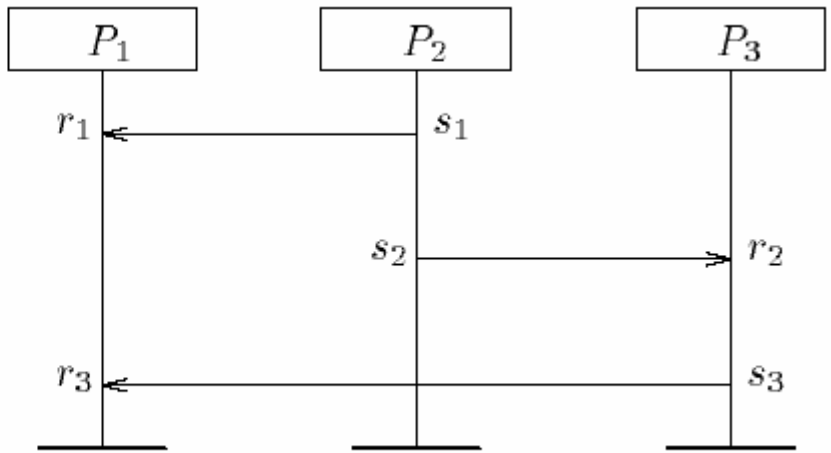
- $H = \{N, B, V^I, V^T, \mu, E\}$, *wobei*
  - N – endliche Knotenmenge
  - B – endliche Menge von Boxen
  - $V^I \in N \cup B$ – Anfangsknoten oder Box
  - $V^T \in N \cup B$ – Endknoten oder Endbox
  - μ – Beschrieftungsfunktion:
    - $\forall$ n $\in N \rightarrow MSC$
    - $\forall$ b $\in B \rightarrow HMSC$ (bereits definierte)
  - $E \subseteq (N \cup B) \times (N \cup B)$ - eine Menge von Kanten

MSC‘s : Anwendung

- werden an früheren Stadien des Systementwurfs angewandt
- Fehlererkennung
- Model-checking
  - Überprüfung von speziellen MSC‘s –Eigenschaften (z.B. race conditions)
  - non-local choice
  - process divergence
  - mittels Automaten, Halbordnungslogik

MSC‘s : race conditions



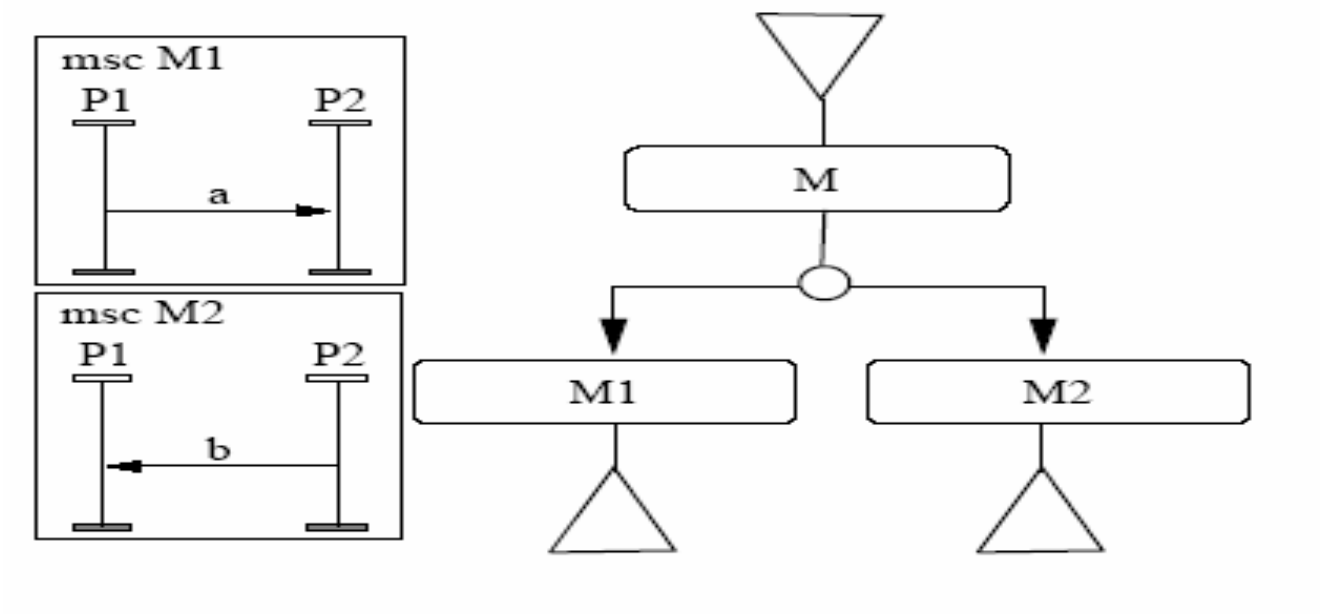


$s_1 < r_1 < s_2 < r_2 < s_3 < r_3$

$s_1 < r_1 < s_2 < r_2 < s_3 < r_3$???

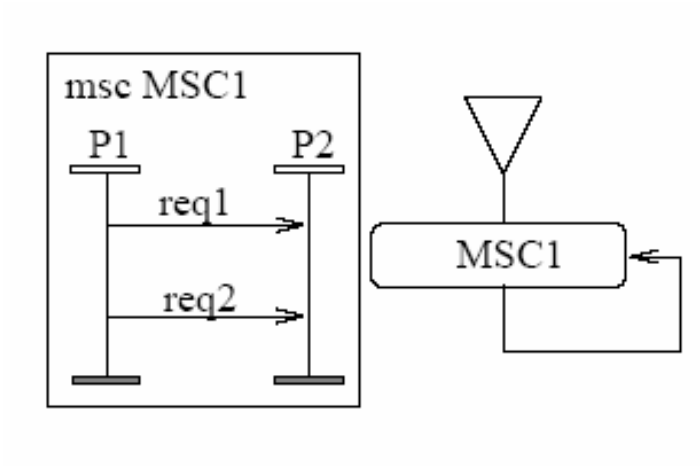
$s_1 < s_2 < r_2 < s_3 < r_3 < r_1$

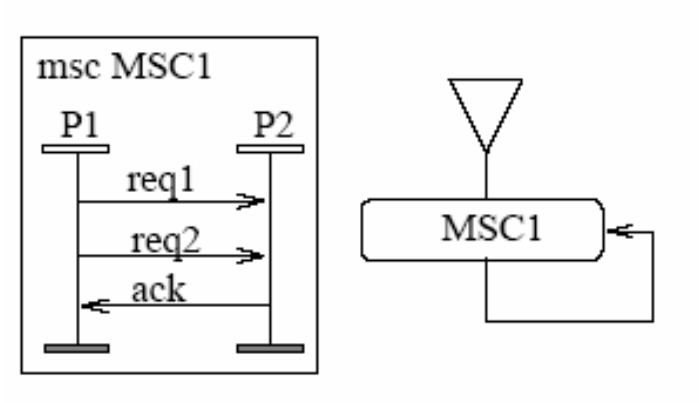
MSC‘s : non-local choice



- Implementierung von einzelnen Prozessen ist nicht trivial
- zusätzliche Information über andere Prozesse notwendig

MSC‘s : process divergence





- keine Information über die Nachrichtenverbindung, Queuestrategie, Prozessorgeschwindigkeit $\Rightarrow$ P_1 kann schneller als P_2 sein
- Folgen:
  - P_2 ist überflutet mit Nachrichten
  - Implementation unterscheidet sich von der Spezifikation (req_1 und req_2 werden überschrieben oder verworfen)

Zusammenfassung

- basic MSC‘s
  - parallele Prozesse
  - asynchrone Kommunikation via Nachrichtenaustauch
- MSG
  - Iterationen, Bedingungen, sequenzielle Ausführung
- HMSC
  - mehrere Hierarchien möglich
- Fehlererkennung in der früheren Stadien der Entwicklung