Institut de Recherche en Informatique de Toulouse
CNRS
INPT
UPS
UT1

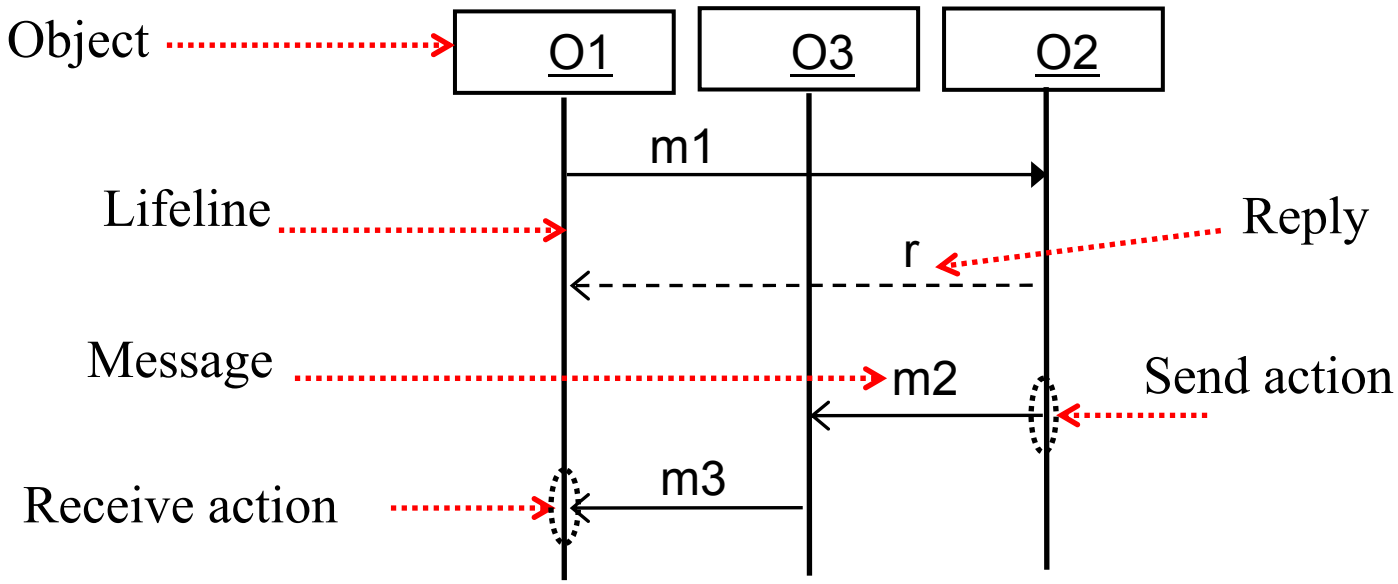
Ambiguity and Structural Properties of Basic Sequence Diagrams

C. Sibertin-blanc
N. Hameurlain
O. Tahir

Presented by J-M Bruel

UML & FM Workshop, Kitakyushu, Japan, 27/10/2008

Basic UML Sequence Diagrams



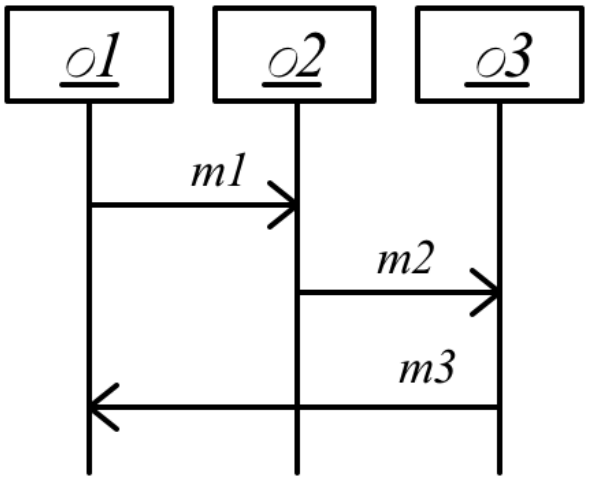
- Change from UML 1 to UML 2 in the definition of SD
  - ✓ the same graphical representation (concrete syntax)
  - ✓ different definitions in the metamodel (abstract syntax)

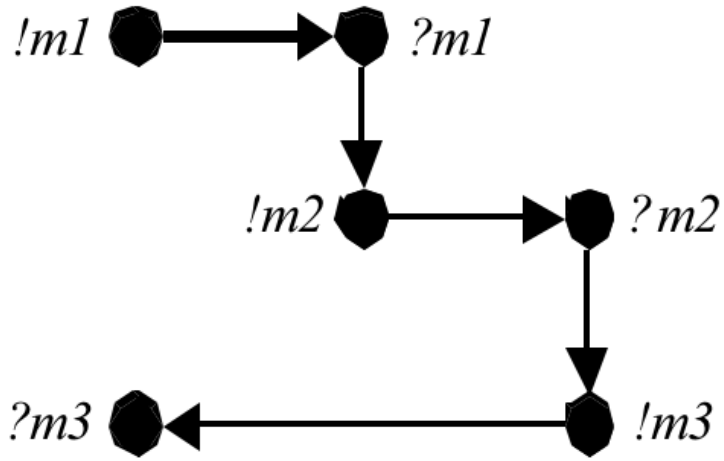
UML 1 or UML 2 Sequence Diagrams?

- UML1 SD : *(O, M, From, To, Precede)*
- UML2 SD : $(O, M, To, From, A, (<_o)_{o \in O})$

- Doesn't matter,
  it is easy to translate a UML1 SD into a UML2 one,
  and vice-versa

A partial order operational semantics

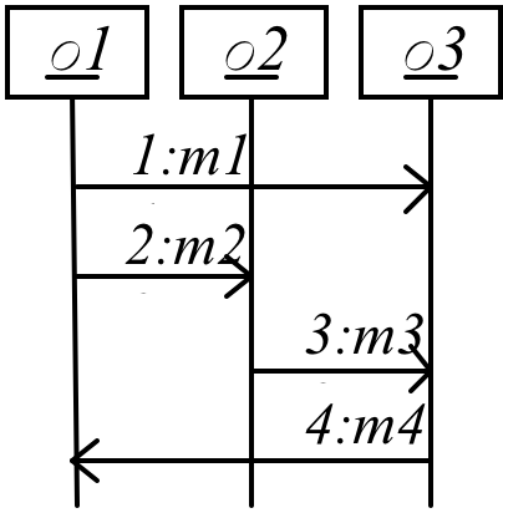


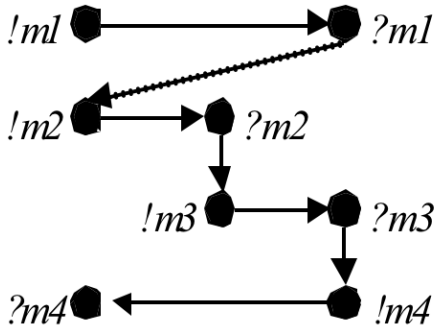


- Actions :
  - !m : send message m
  - ?m : receive message m
- Semantics :

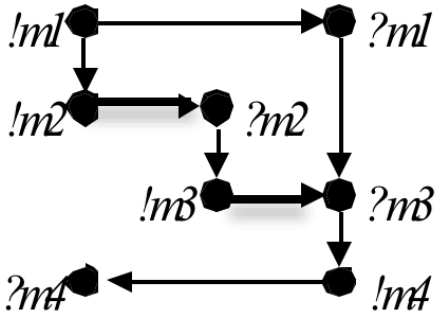
  an action may occur iff the previous ones have already occurred

Various semantics of a simple SD

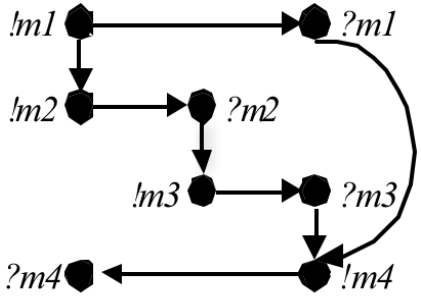




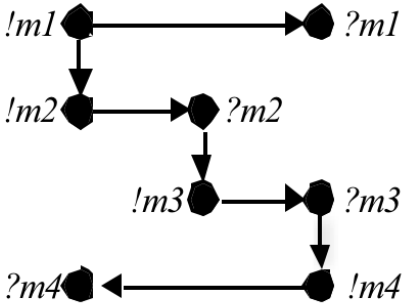
Linear semantics



UML2 semantics



Causal semantics



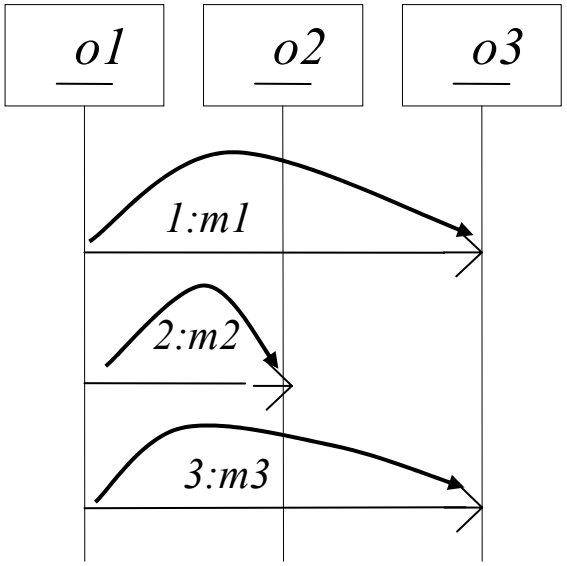
Emission semantics

The synchronization ordering constraints

- Synchronizations between objects:

  the sending of a message always precedes its reception

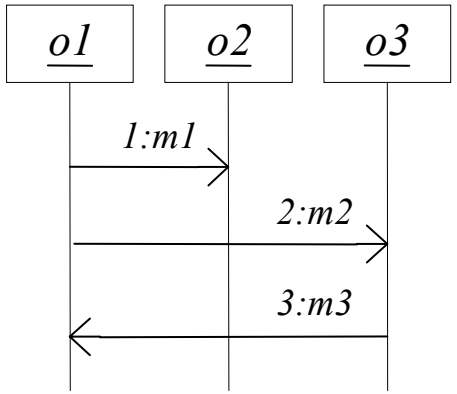
- $<_{SYNC} = \{(!m, ?m) ; m \in M\}$
- These constraints are a part of any semantics

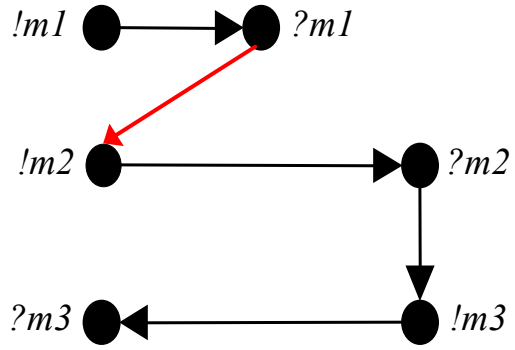


The Linear Semantics

- A message may be sent when all its predecessors are already received: **m1 Precede m2 ==> ?m1 < !m2**
- $<_{re} = \{(?m, !m'); m, m' \in M \text{ and } m \textit{ Precede } m'\}$.

  $<_{\mathbf{linear}} = <_{\mathbf{sync}} \cup <_{\mathbf{re}}$
- The execution of any SD is fully sequential process,





An additional GeneralOrdering constraint is needed

The Emission Semantics

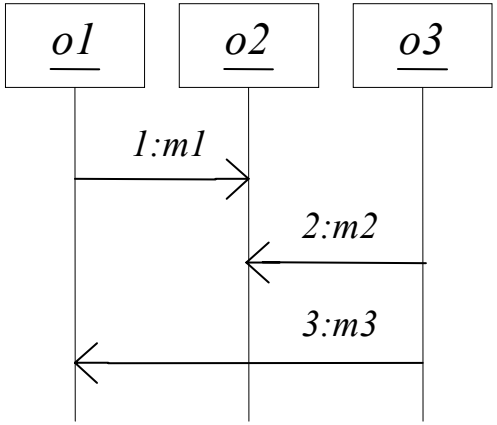
- A message may be sent when all its predecessors are sent

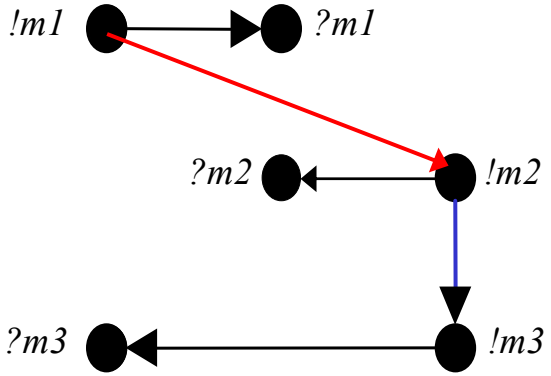
  m1 *Precede* m2 ==> !m1 < !m2

- $<_{\text{emission, re}} = \{(?m, !m'); m, m' \in M, m \textit{ Precede } m' \text{ and } \textit{To}(m) = \textit{From}(m')\}$

  $<_{\text{emission, ee}} = \{(!m, !m'); m, m' \in M, m \textit{ Precede } m' \text{ and } \textit{To}(m) \neq \textit{From}(m')\}$

$$<_{\text{emission}} = <_{\text{sync}} \cup <_{\text{emission, ee}} \cup <_{\text{emission, re}}$$





A GeneralOrdering constraint is still needed

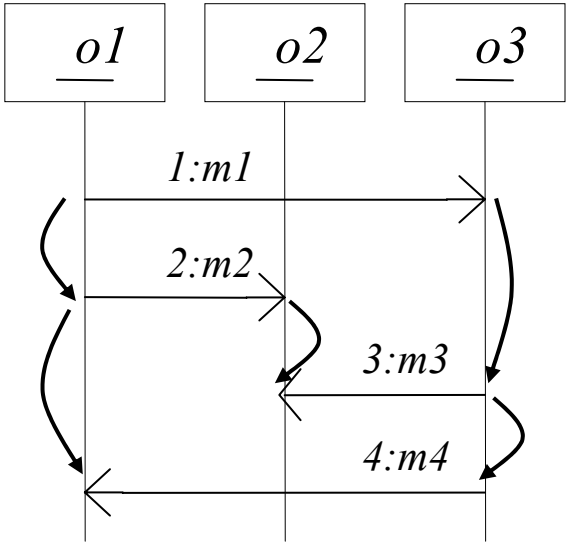
The UML2 Semantics

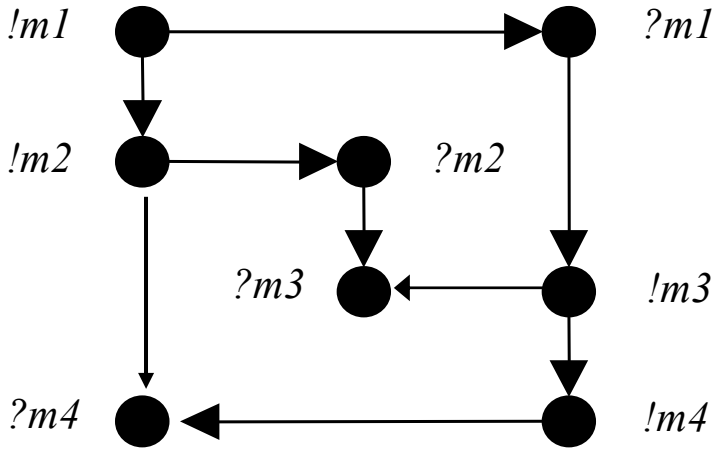
- Each object performs actions in sequence,

  the only synchronizations between objects are the message exchanges.

- $<_{\mathbf{local}} = \{(a, a');\ a, a' \in A,$ *executed by the same object* and
  msg(a) Precede msg(a')$\}$

$$<_{\mathbf{MSC}} = <_{\mathbf{sync}} \cup <_{\mathbf{local}}$$



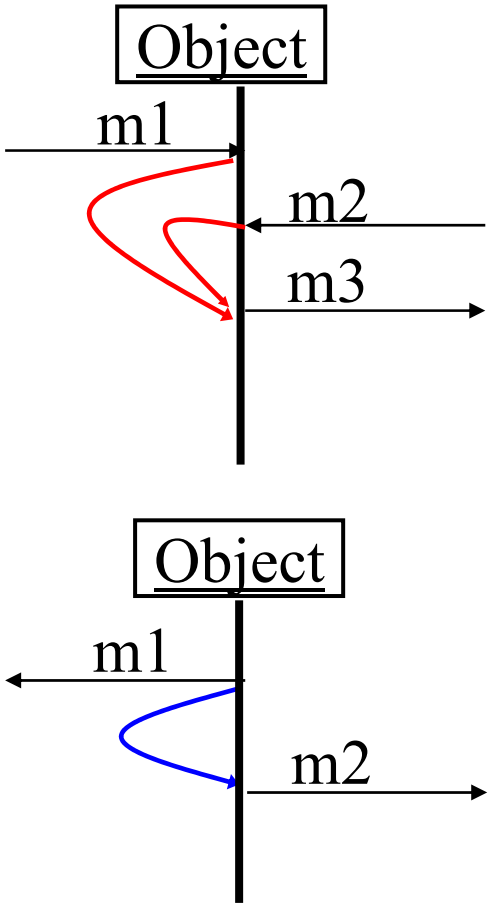


The Causal Semantics (1)

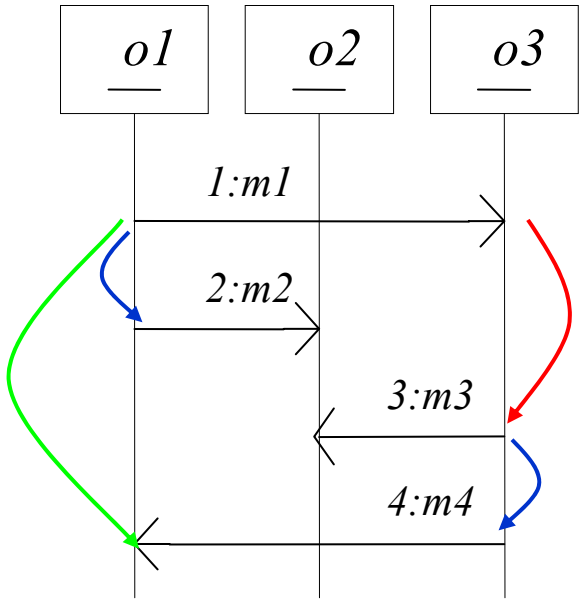
Weaken the MSC-semantics

- The reception of a message has no local cause
- A send action is caused
  - by the previous receive actions
    $<_{causal,\ re} = \{(?m1, !m3), (?m2, !m3)\}$
  - or by the previous send action
    $<_{causal,\ ee} = \{(!m1, !m2)$

- $<_{causal} = <_{sync} \cup <_{causal,\ re} \cup <_{causal,\ ee,}$

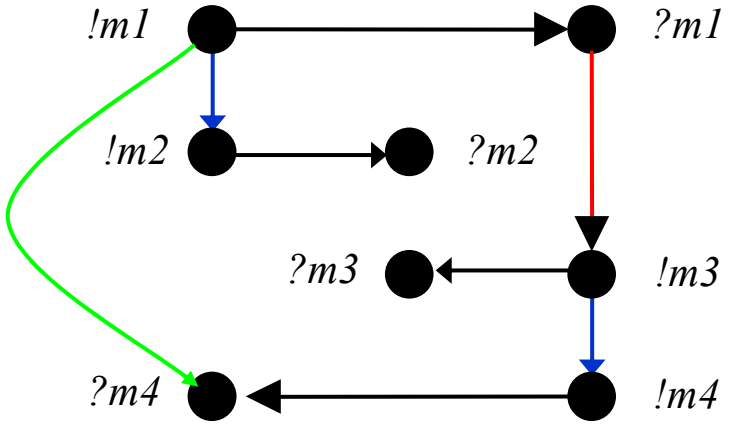


The Causal Semantics (2)



$<_{causal,\ re} = \{(?m1, !m3)\}$

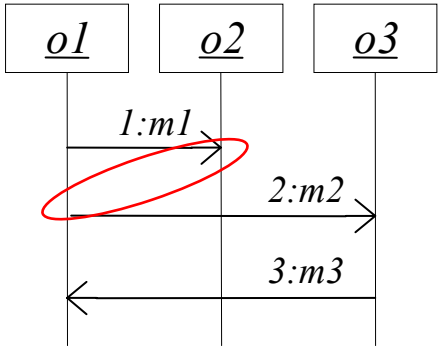
$<_{causal,\ ee} = \{(!m1, !m2), (!m3, !m4)\}$



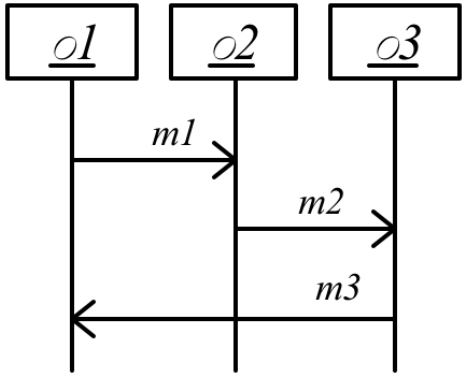
This semantics avoids unnecessary ordering constraints at the specification level

Sequential SDs

- A SD is **sequential** iff any message is sent
  by the object that has received the previous
  message.



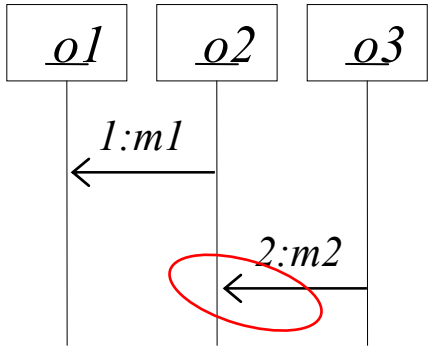
- Th : A SD is sequential iff
  $<_{\mathbf{emission}} \approx <_{\mathbf{causal}} \approx <_{\mathbf{UML2}} \approx <_{\mathbf{lineaar}}$



- All the semantics of a sequential SD are the same, it suffers no ambiguity

Locally Causal SDs

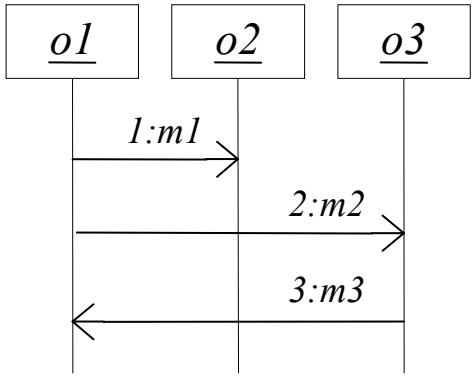
- A SD is **Locally Causal** if any reception in an object is the transitive consequence of a send action



- Th : A SD is locally causal iff

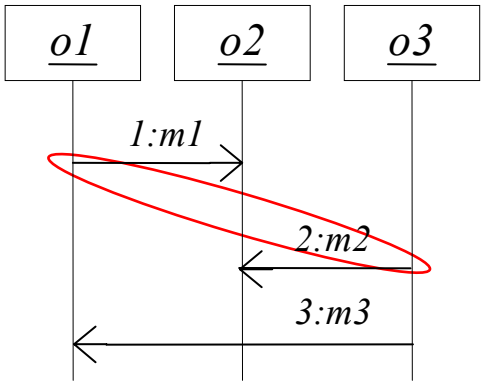
$<_{\mathbf{emission}} \approx <_{\mathbf{causal}} \approx <_{\mathbf{UML2}} \subseteq <_{\mathbf{lineaar}}$

A locally causal SD support 2 different semantics



Locally controlled SDs

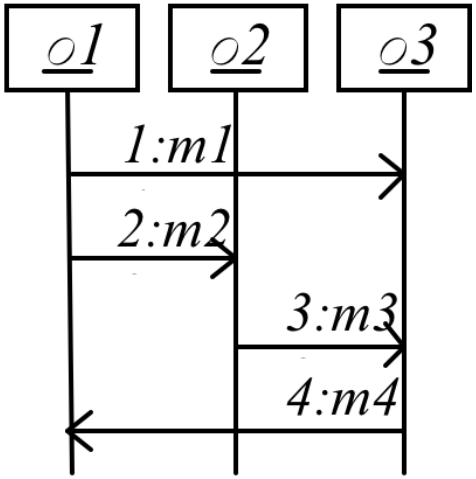
- A SD is **Locally Controlled** iff any message is sent by the object that has either sent or received the previous message.



- Th : A SD is locally controlled iff

$<_{\mathbf{emission}} \subset_C <_{\mathbf{causal}} \subset_C <_{\mathbf{UML2}} \subset_C <_{\mathbf{linear}}$

A locally controlled SD supports 4 different semantics



Relationships Between the Properties

- ✓ None of these properties:

  $<_{\text{emission}} \subseteq <_{\text{linear}}$

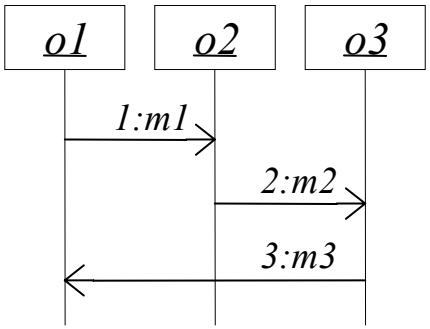
  $<_{\text{causal}} \subset_C <_{\text{UML2}} \subset_C <_{\text{linear}}$

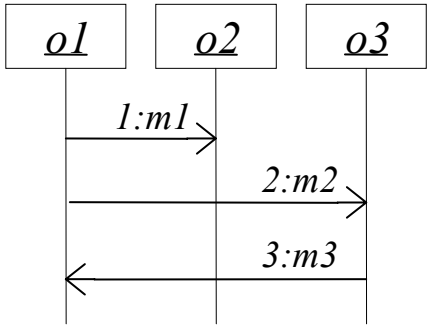
  The four semantics differ by the ordering constraints between actions, the level and type of concurrency

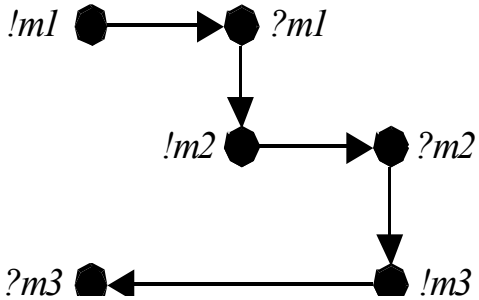
- ✓ Sequential ==> Locally Causal ==> Locally Controlled

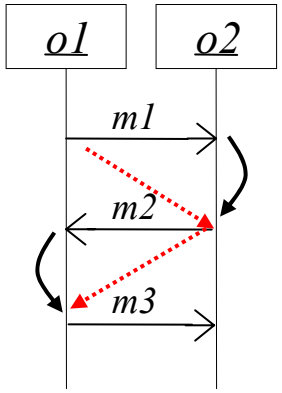
Conclusion

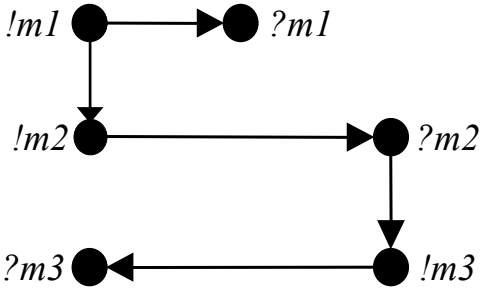
- Basic SDs have a visually appealing graphical representation that can go with semantic ambiguities
- The level of ambiguity of a basic SD, that is the differences between its different semantics, depends on the fulfilment of some structural properties
- The more a SD is ambiguous, the more its validation is important


o1
o2
o3
1:m1
2:m2
3:m3


o1
o2
o3
1:m1
2:m2
3:m3


!m1
?m1
!m2
?m2
?m3
!m3


o1
o2
m1
m2
m3


!m1
?m1
!m2
?m2
?m3
!m3