From Interaction Overview Diagrams to PEPA nets

Leïla Kloul
PRiSM, Université de Versailles

Juliana Küster-Filipe
School of Computer Science
The University of Birmingham

Contents

1. Motivation
2. Design Notation
   - Dynamic Models in UML 2.0
   - IODs vs Activity Diagrams
   - Using IODs for Mobility
3. PEPA nets
4. Formal Translation
5. Conclusions

Contents

Motivation

- Model and analyse mobile systems.
- System consists of several locations. Some objects can move between locations. Within a location objects can interact with others.
- Choice of modelling language: for designers vs for analysis.

Motivation

- Model and analyse mobile systems.
- System consists of several locations. Some objects can move between locations. Within a location objects can interact with others.
- Choice of modelling language: for designers vs for analysis.

UML 2.0 vs PEPA nets

Dynamic Models in UML 2.0

- Individual behaviour is modelled using state diagrams - intra-object behaviour.
- Collaborative behaviour is modelled using interaction diagrams - inter-object behaviour. These diagrams include
  - Sequence diagrams
  - Interaction overview diagrams (IODs)

Sequence diagrams

- Are a visual scenario-based formalism.
- Describe the instances involved in an interaction, and the messages exchanged for the interaction (partially ordered over time).
- Contain two dimensions: a vertical dimension denoting time; a horizontal dimension representing the instances involved in the interaction.

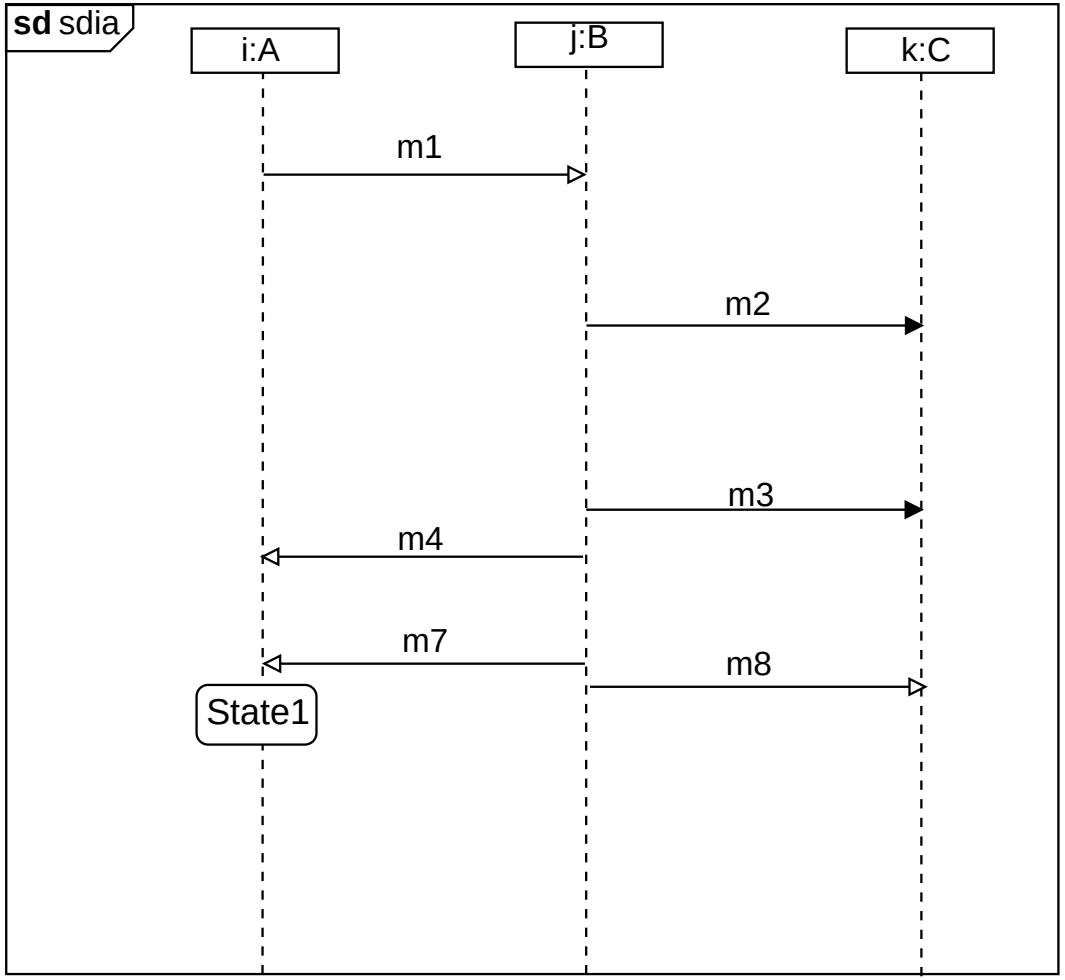
Sequence diagrams in UML 2.0

- Have new improved structure and expressiveness (through so-called *interaction fragments*)
- Loops and conditional branching can be indicated more clearly
- Parallel behaviour can be expressed as well as ordering of non-related communications

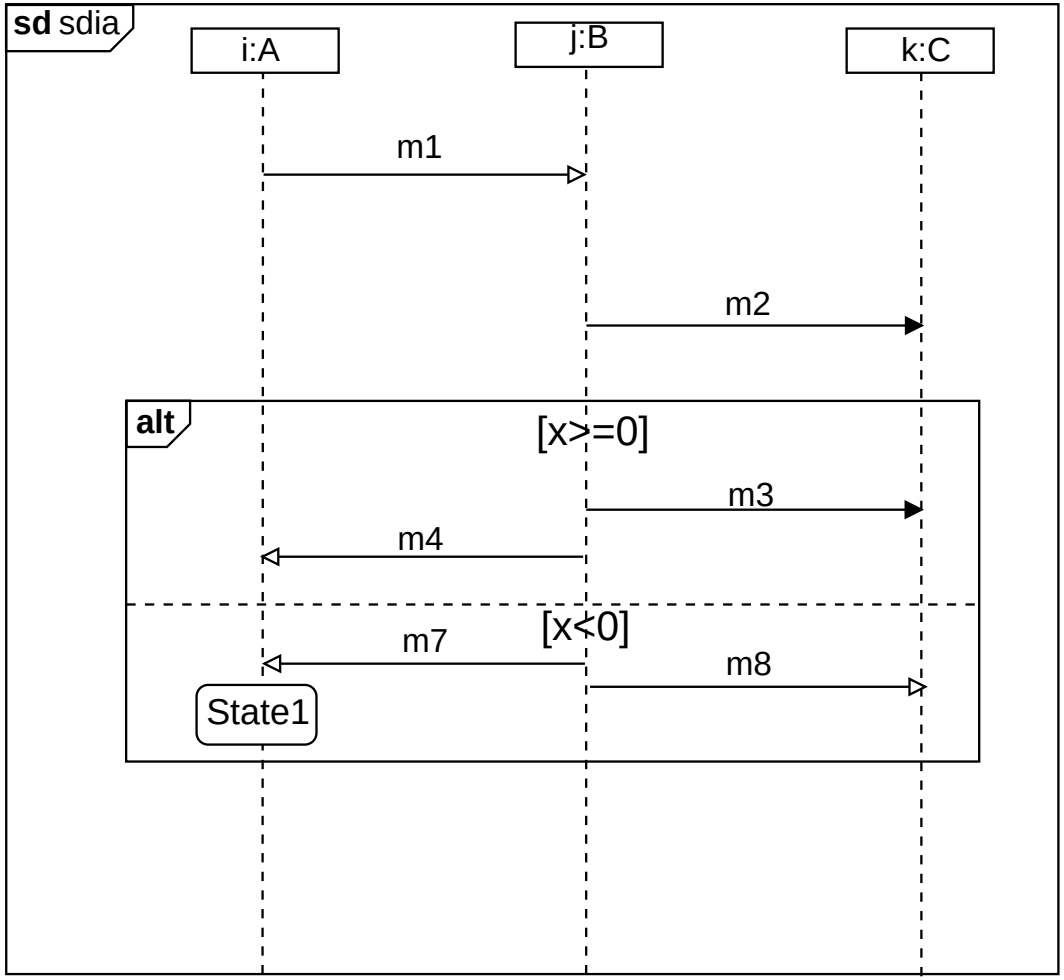
Sequence diagrams in UML 2.0

- Have new improved structure and expressiveness (through so-called *interaction fragments*)
- Loops and conditional branching can be indicated more clearly **loop** **alt**
- Parallel behaviour can be expressed as well as ordering of non-related communications **par** **strict**

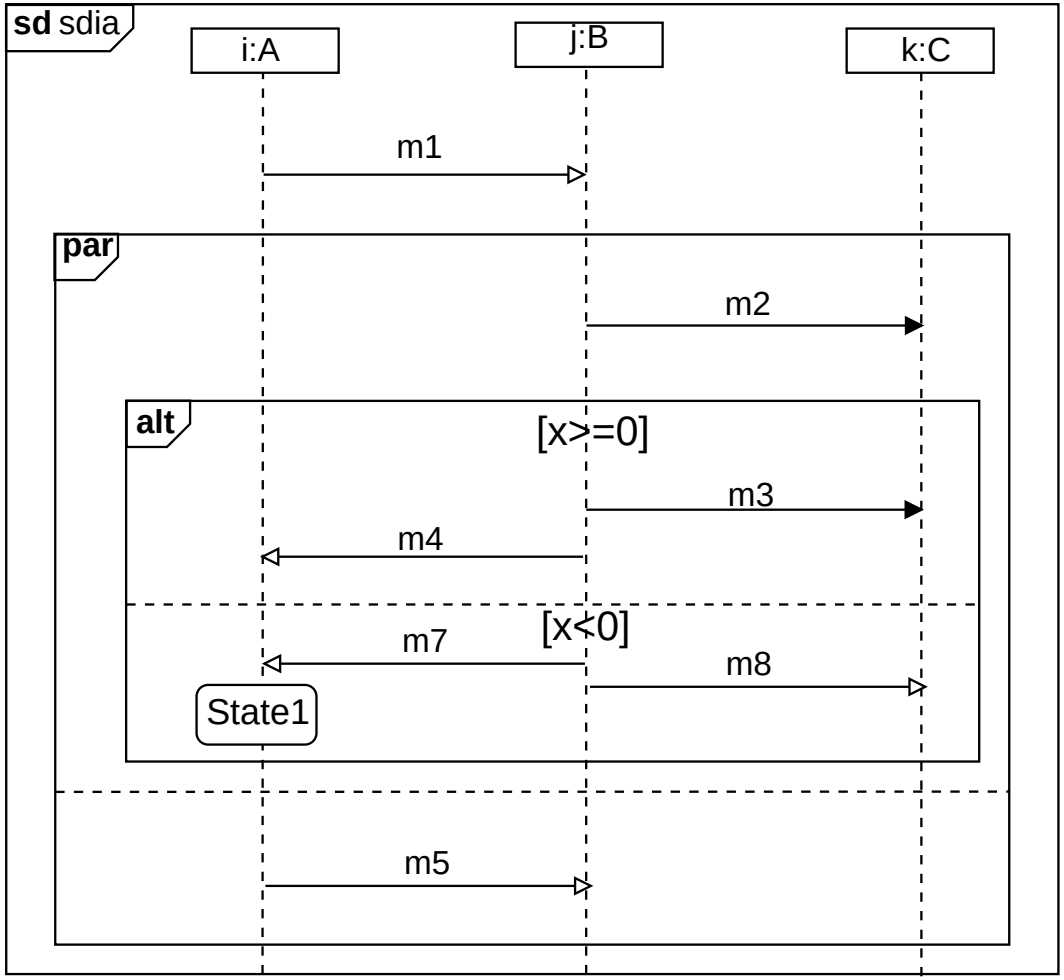
Example Diagram



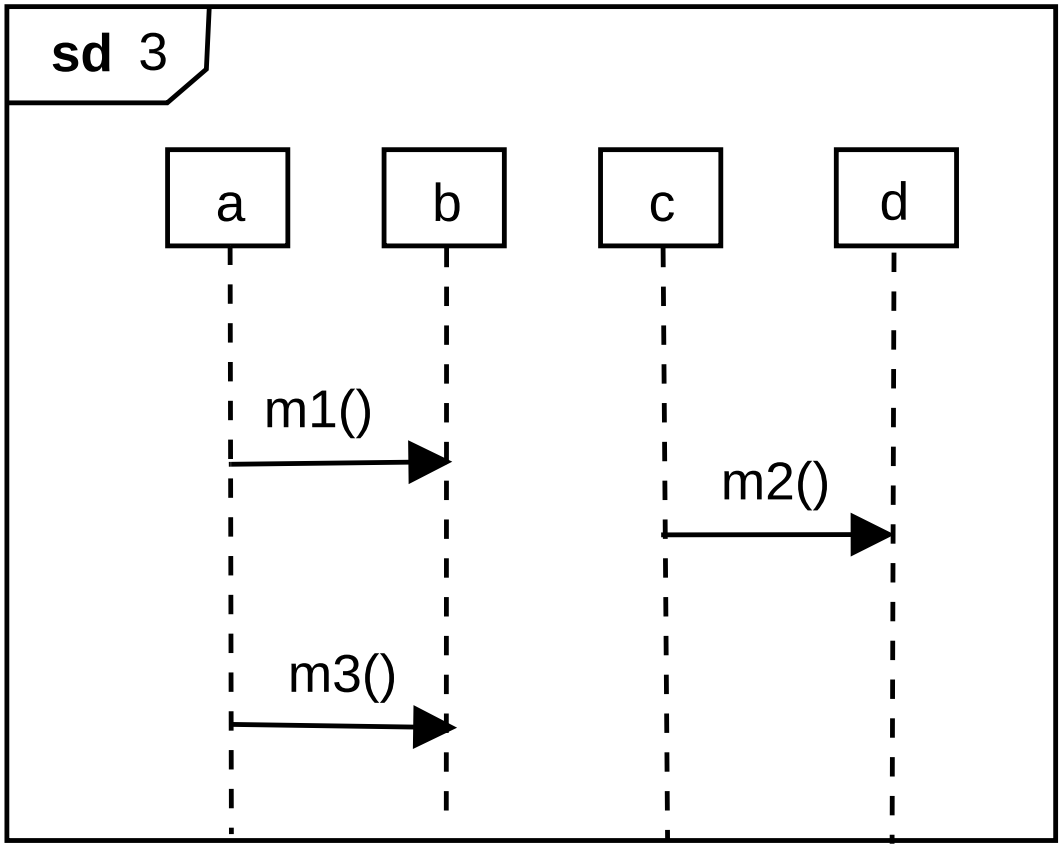
Example Diagram



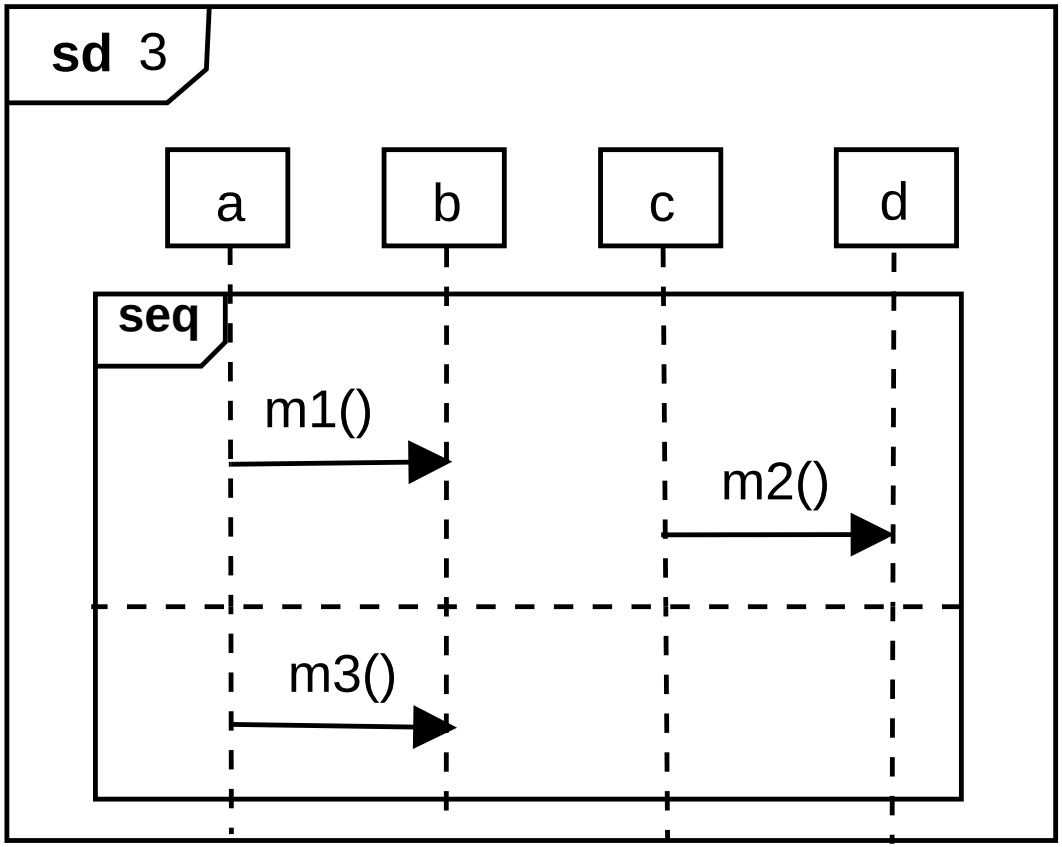
Example Diagram



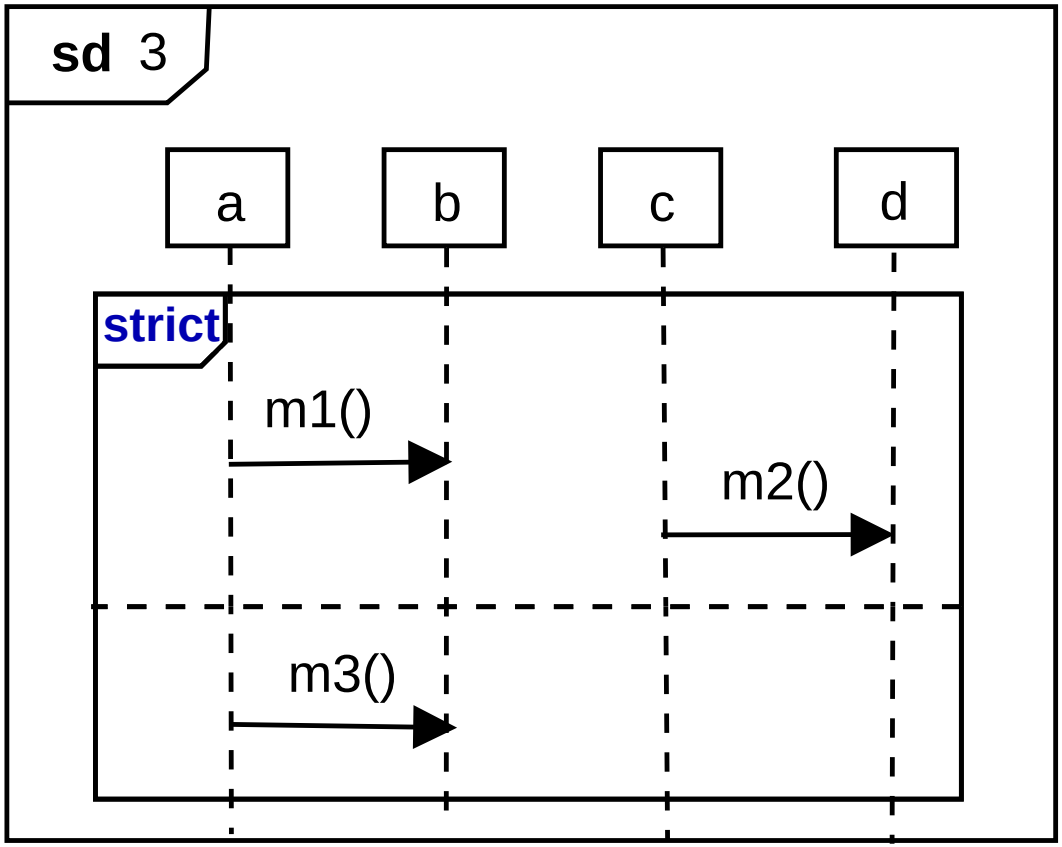
Ordering



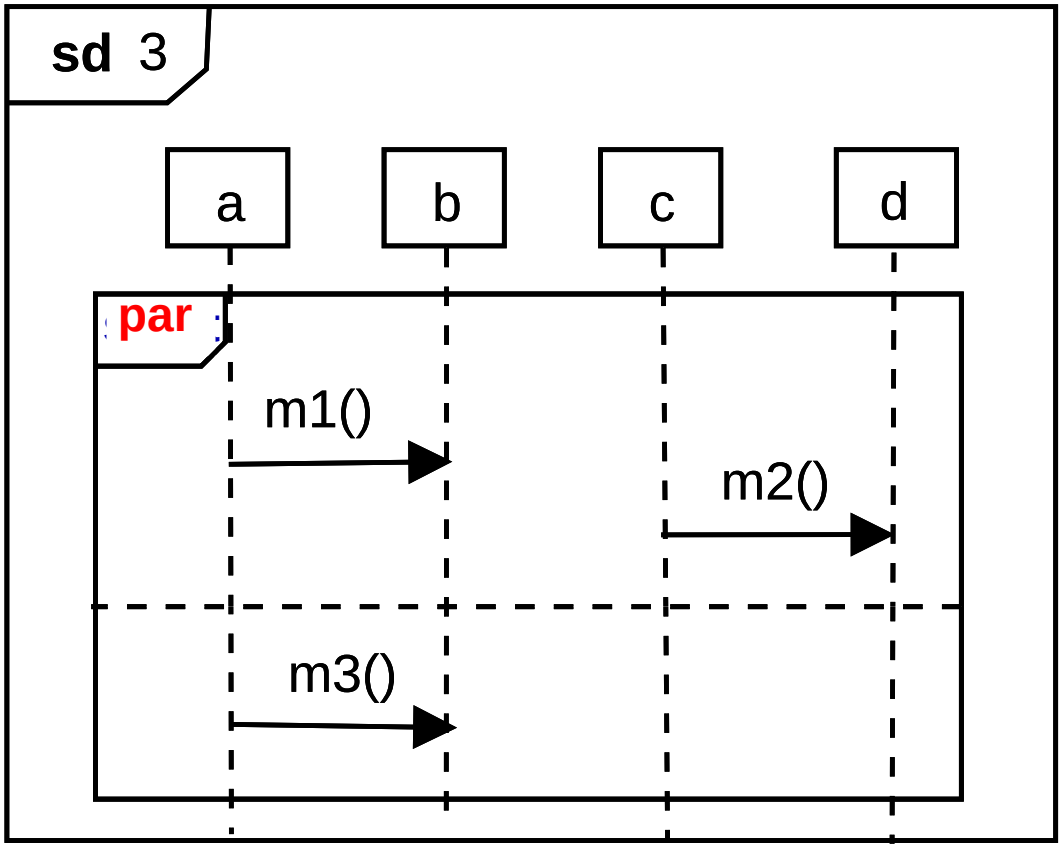
Ordering



Ordering

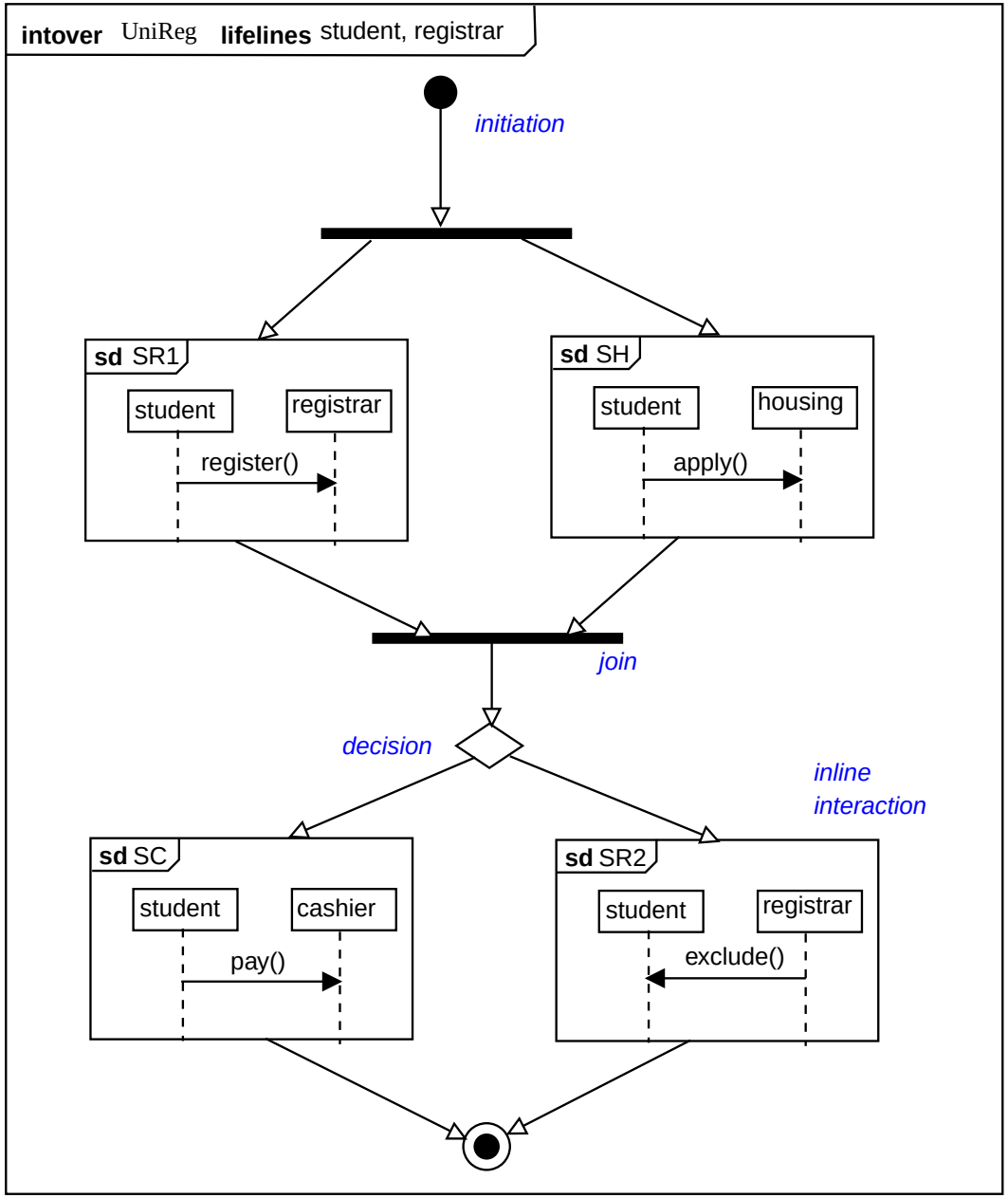


Ordering

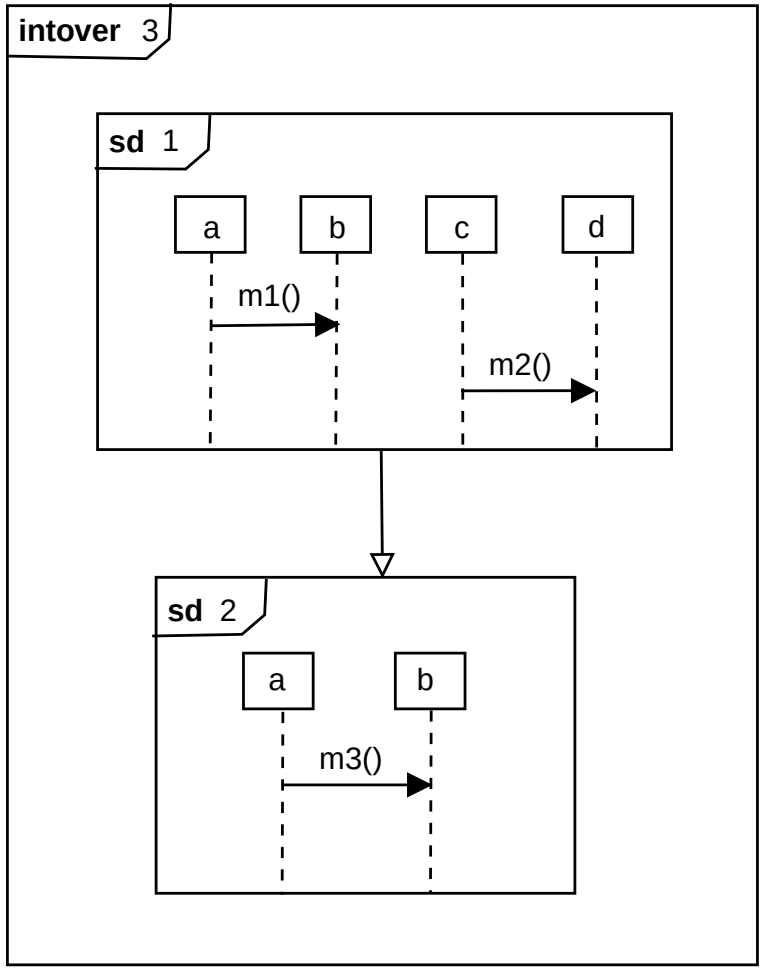


Interaction Overview Diagram

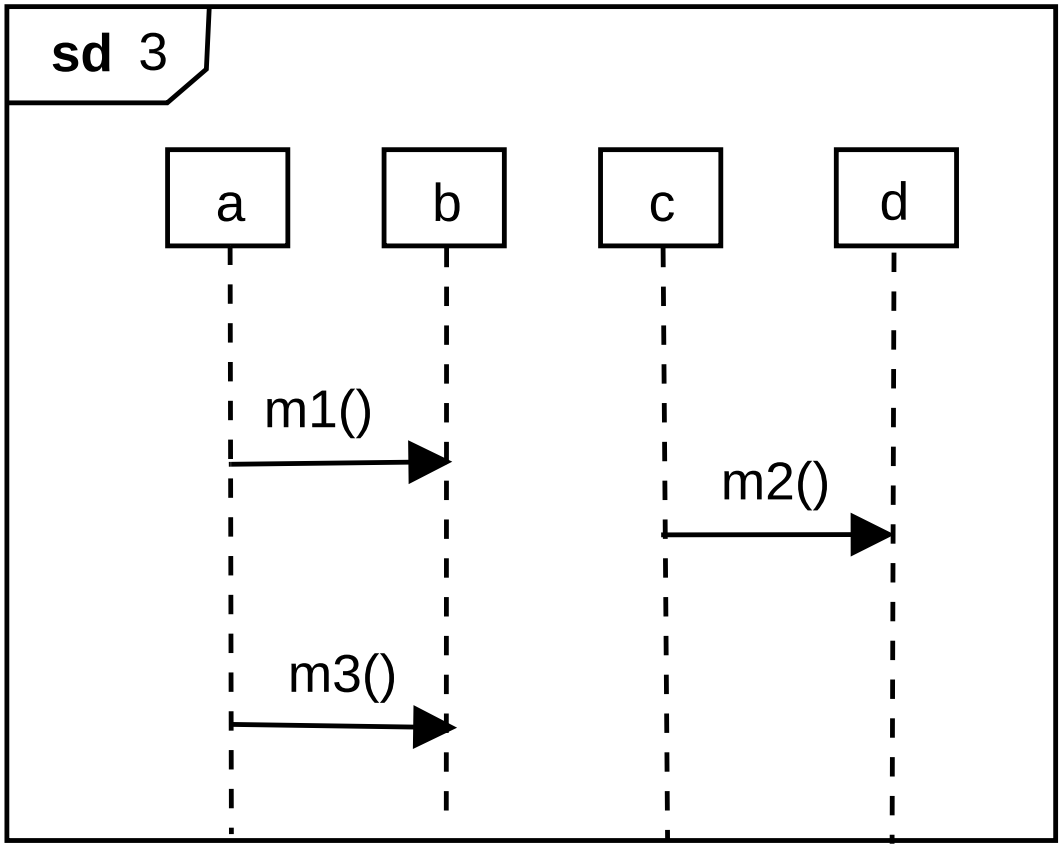
- High level structuring mechanism for sequence diagrams.
- Special kind of activity diagram with control flow only.
- Nodes are interactions (sequence diagrams); edges show the order in which these interactions occur.
- Uses forks, joins, decision and merge nodes from activity diagrams.


intover UniReg lifelines student, registrar
initiation
sd SR1
student
registrar
register()
sd SH
student
housing
apply()
join
decision
inline
interaction
sd SC
student
cashier
pay()
sd SR2
student
registrar
exclude()

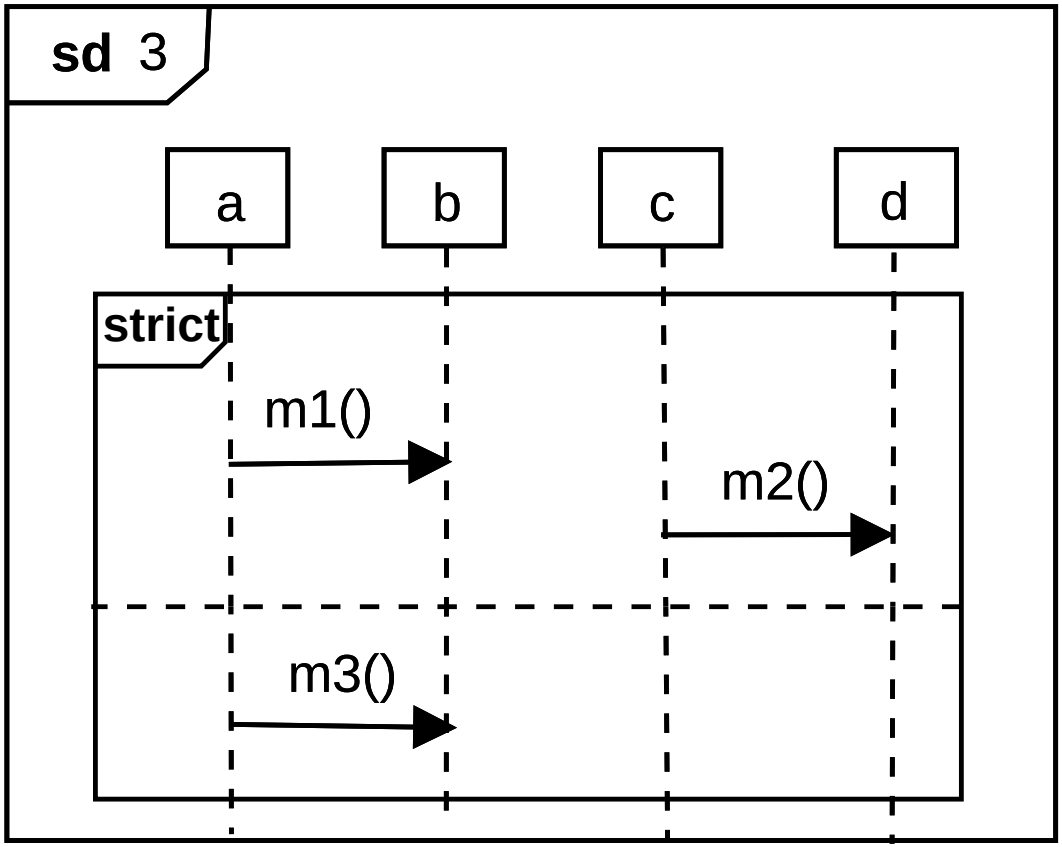
Semantics of an IOD



Flattening



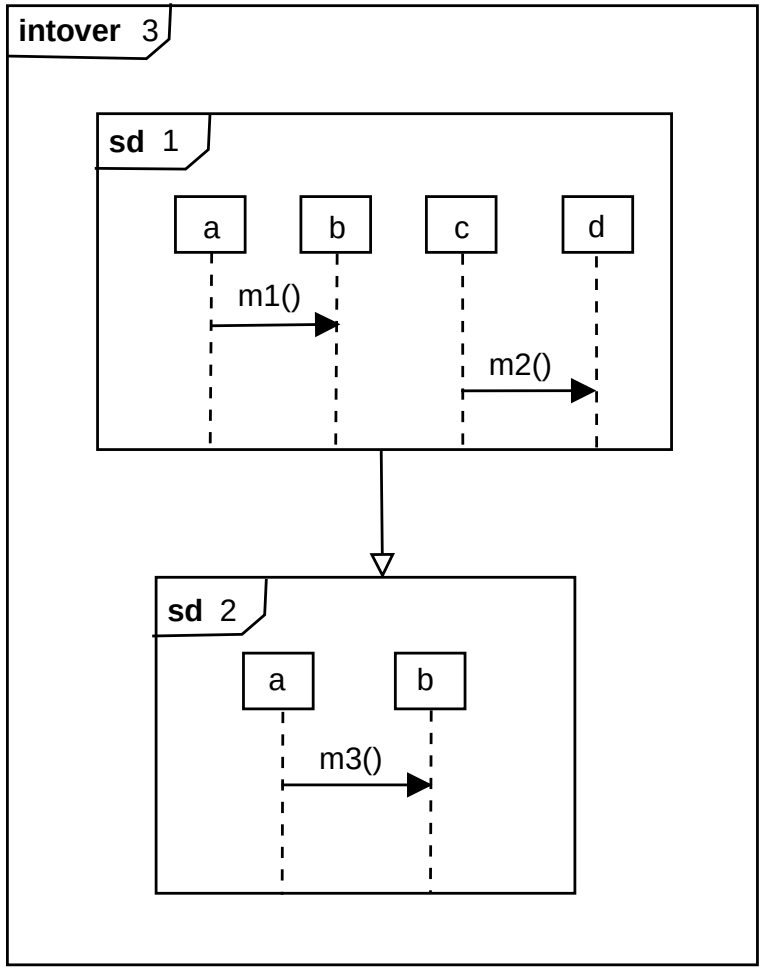
Flattening



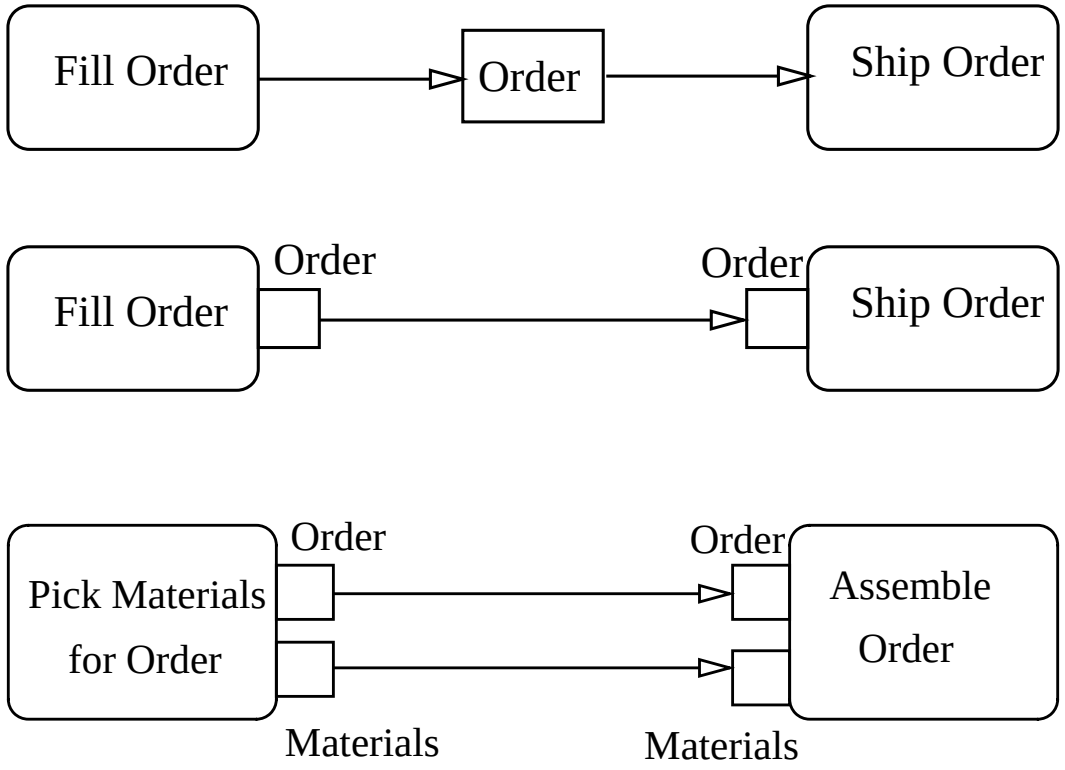
Which Case to Allow?

- Both interpretations are sensible: strong and weak sequential composition.
- We need different notation for each case.

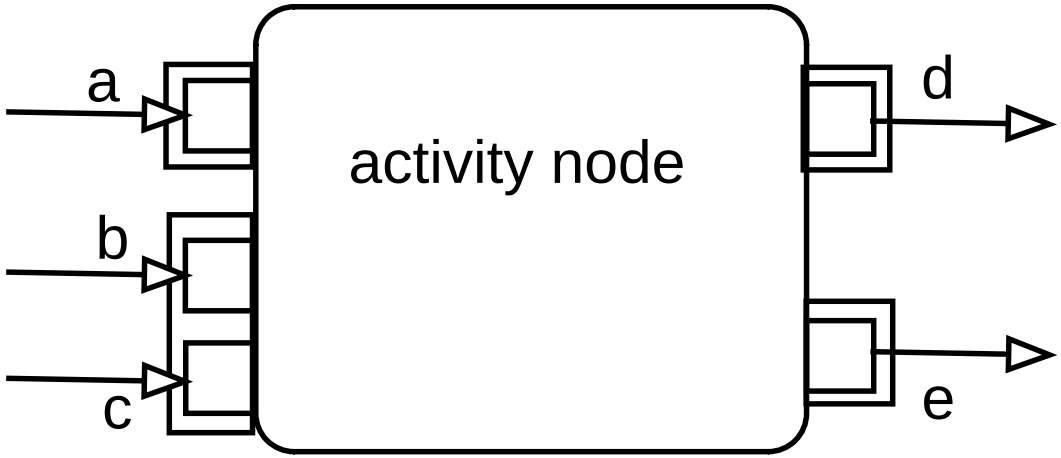
Strong Sequential Composition



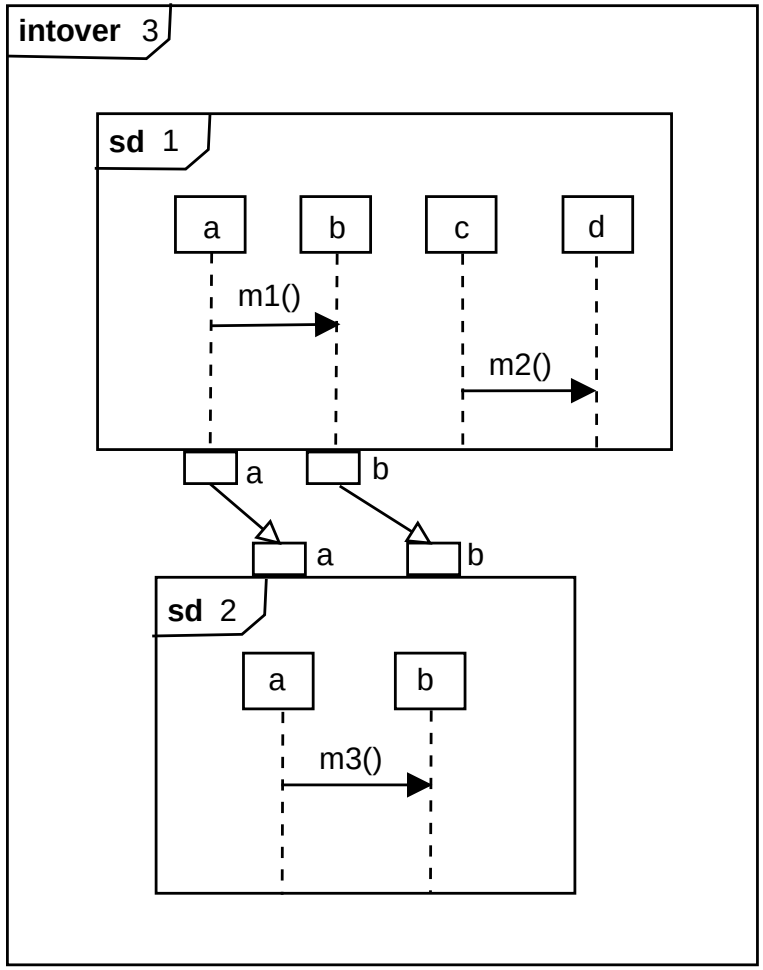
Object Flow in ADs



Alternative input/output pins



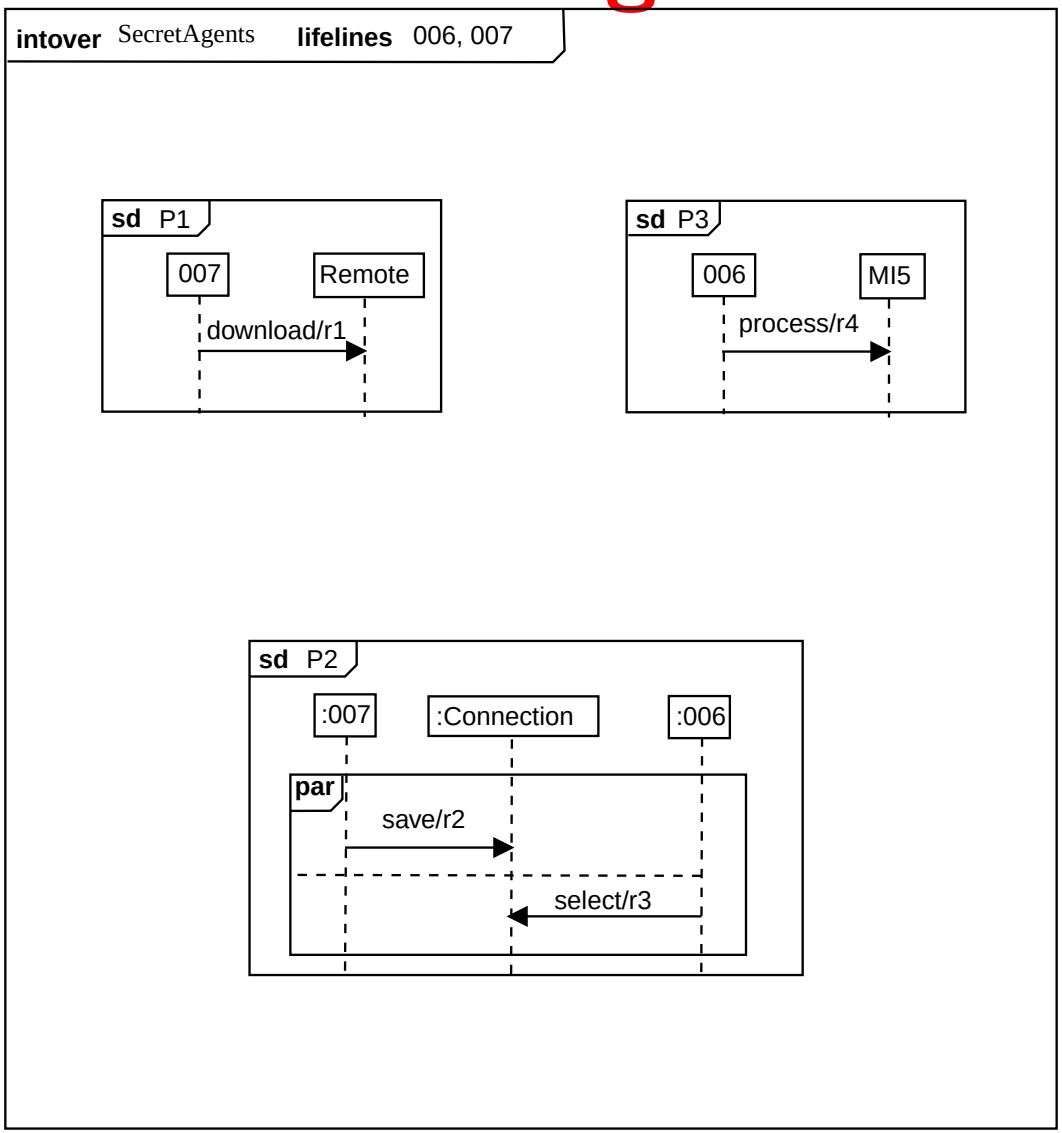
Weak Sequential Composition



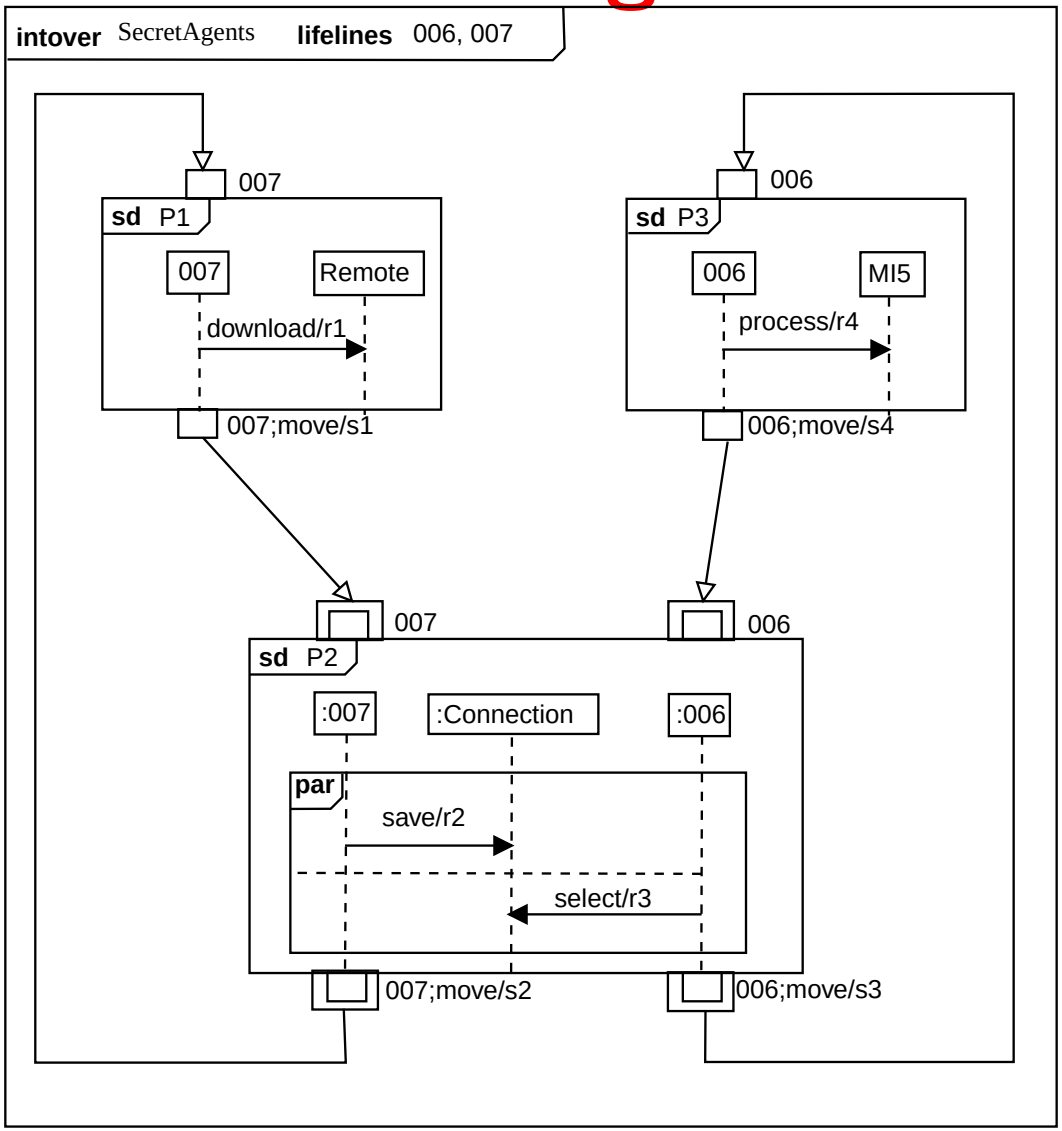
IODs for Mobility

- Nodes represent locations in the system (interaction name).
- Explicit object flow is used to indicate object mobility.
- $\{upperBound = value\}$ indicates the maximum number of tokens allowed in a pin.
- $\{initBound = value\}$ indicates the initial number of tokens in a pin.
- $\{weight = value\}$ indicates the number of tokens carried in an edge.

Secret Agents



Secret Agents



Secret Agents