

# From Interaction Overview Diagrams to PEPA nets



Leila Kloul

*Join work with J. Küster-Filipe*

## ■ From IODs to PEPA nets

UML 2

PEPA nets

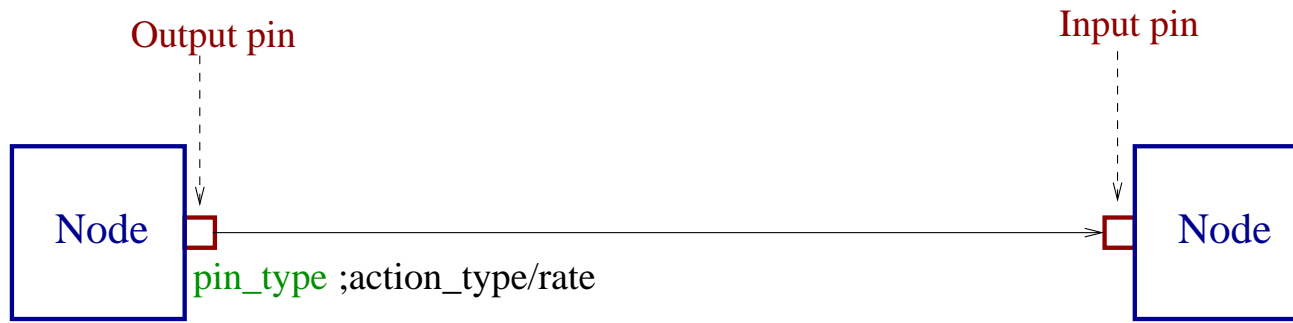
- IOD → PEPA net
- IOD node → Place
- IOD edge → Firing transition
- Object inside a node → PEPA component

- How to obtain PEPA components

- **Static component:** Object in the IOD node (non dynamic) defined by its *name* (optional) and its *type*

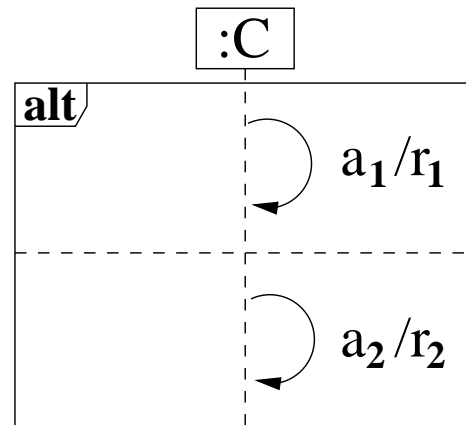
*name* : *type*

- **Token:** Dynamic object in the IOD node. The token *type* is obtained from the *pin type*.



- How to obtain PEPA activities

- Individual activities: these are *messages* sent by an object to itself.



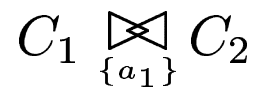
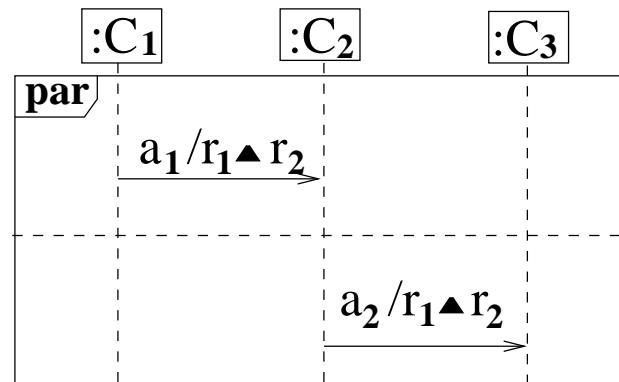
$$C \stackrel{def}{=} (a_1, r_1).C' + (a_2, r_2).C''$$

- Firing transition: the activity of a firing transition is extracted from the label on the *output pin* of the corresponding IOD edge.

*pin\_type; action\_type/rate*

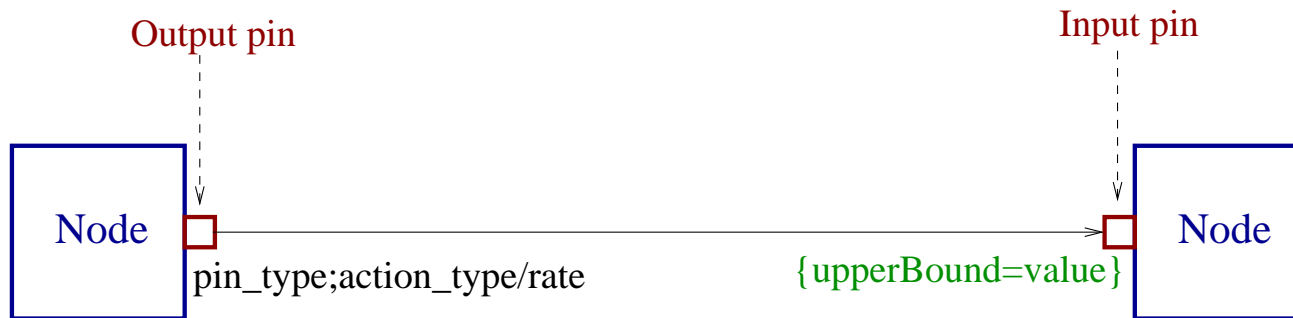
- How to obtain PEPA activities

- Cooperation activities: these are the *messages* involving more than one object in an IOD node.



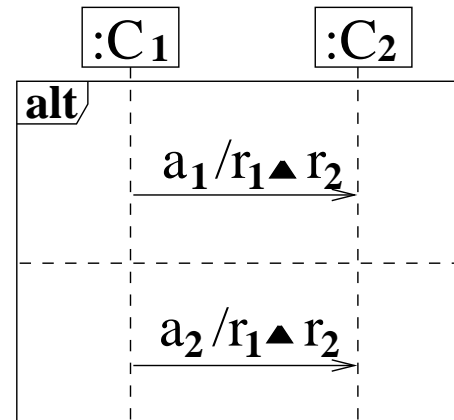
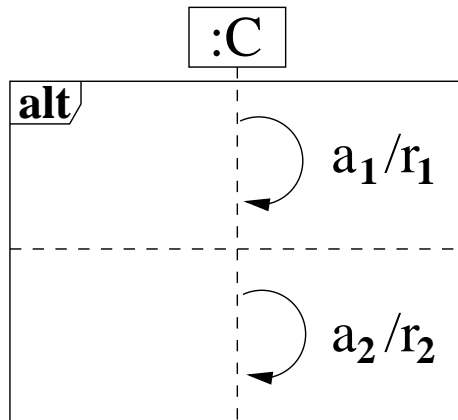
- How to obtain the number of cells

- The number of cells is given by the **value** of parameter *upperBound* on the *input pin* label.



- Translating the UML interaction operators

- Operator *alt*



$$C \stackrel{def}{=} (a_1, r_1).C' + (a_2, r_2).C''$$

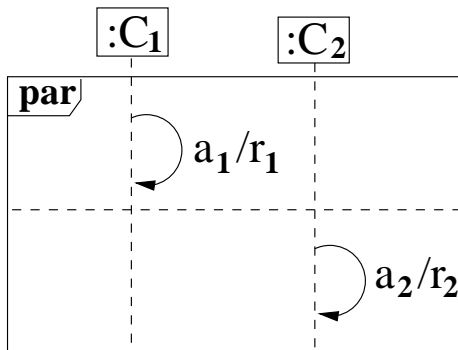
$$C_1 \stackrel{def}{=} (a_1, r_1).C'_1 + (a_2, r_1).C''_1$$

$$C_2 \stackrel{def}{=} (a_1, r_2).C'_2 + (a_2, r_2).C''_2$$

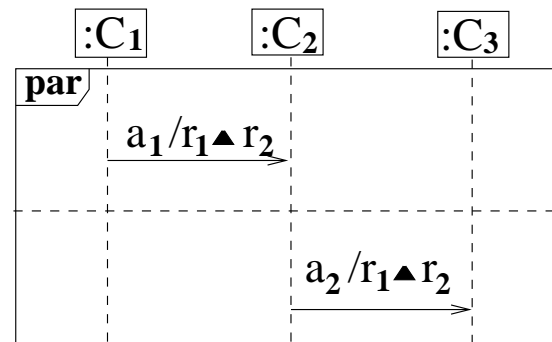
$$C_1 \boxtimes_{\{a_1, a_2\}} C_2$$

- Translating the UML interaction operators

- Operator *par*



$$(a_1, r_1).C'_1 \parallel (a_2, r_2).C'_2$$

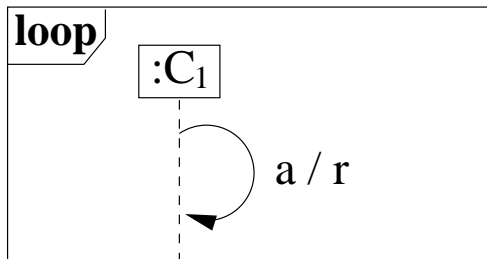


$$C_1 \begin{matrix} \boxtimes \\ \{a_1\} \end{matrix} C_2 \parallel C_2 \begin{matrix} \boxtimes \\ \{a_2\} \end{matrix} C_3$$

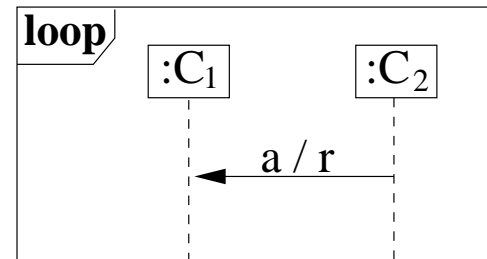


- Translating the UML interaction operators

- Operator *loop*



$$C_1 \stackrel{def}{=} (a, r).C_1$$

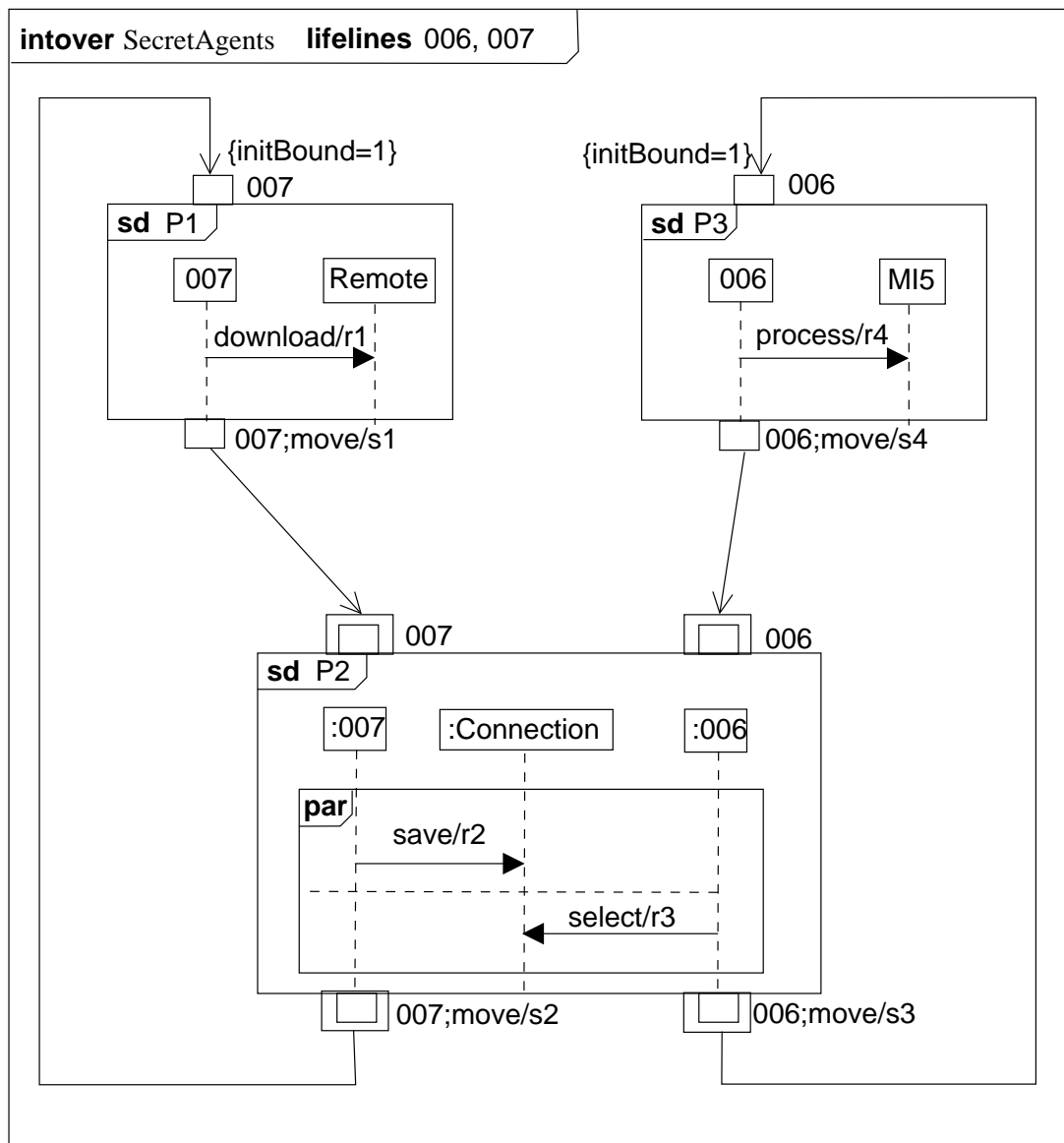


$$C_1 \stackrel{def}{=} (a, \top).C_1$$

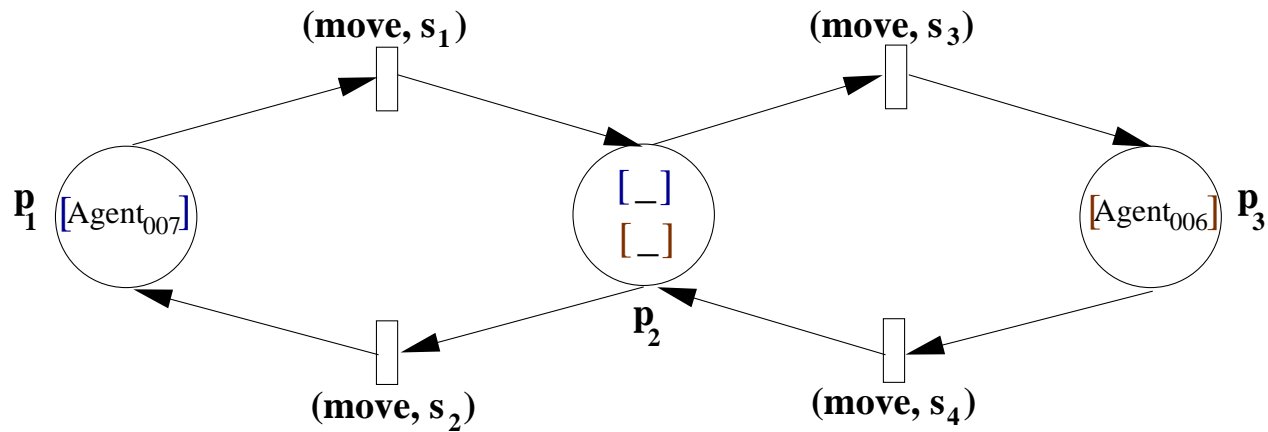
$$C_2 \stackrel{def}{=} (a, r).C_2$$

$$C_1 \boxtimes_{\{a\}} C_2$$

# ■ Example



## ■ Example



- Static components

*Remote*  $\stackrel{\text{def}}{=} (\text{download}, \top). \text{Remote}$

*Connection*  $\stackrel{\text{def}}{=} (\text{save}, \top). \text{Connection} + (\text{select}, \top). \text{Connection}$

*MI5*  $\stackrel{\text{def}}{=} (\text{process}, \top). \text{MI5}$

## ■ Example

- Components tokens

$$Agent_{007} \stackrel{def}{=} (download, r_1).Agent'_{007} + (save, r_2).Agent'_{007}$$

$$Agent'_{007} \stackrel{def}{=} (\mathbf{move}, \top).Agent_{007}$$

$$Agent_{006} \stackrel{def}{=} (select, r_3).Agent'_{006} + (process, r_4).Agent'_{006}$$

$$Agent'_{006} \stackrel{def}{=} (\mathbf{move}, \top).Agent_{006}$$

- Places

$$P_1 \stackrel{def}{=} Remote \quad \begin{array}{c} \boxtimes \\ \{download\} \end{array} \quad Agent_{007}[Agent_{007}]$$

$$P_2 \stackrel{def}{=} Connection \quad \begin{array}{c} \boxtimes \\ \{save, select\} \end{array} \quad (Agent_{007}[-] \parallel Agent_{006}[-])$$

$$P_3 \stackrel{def}{=} MI5 \quad \begin{array}{c} \boxtimes \\ \{process\} \end{array} \quad Agent_{006}[Agent_{006}]$$

## ■ Future Work

- to be implemented ...
- to be extended ...