

# **A Unidirectional Transition Fusion for Coloured Petri Nets and its Implementation for the CPNTools**

**João Paulo Barros<sup>1,2</sup>**

`jpb@uninova.pt`

**Luís Gomes<sup>1</sup>**

`lugo@uninova.pt`

<sup>1</sup>Universidade Nova de Lisboa  
Faculdade de Ciências e Tecnologia  
Dep. de Engenharia Electrotécnica  
Campus da FCT-UNL  
2825 Monte de Caparica

<sup>2</sup>Instituto Politécnico de Beja  
Escola Superior de Tecnologia e Gestão  
Área Departamental de Engenharia  
Rua Afonso III, n.º 1  
7800-050 Beja

PORTUGAL

## **CPNs and objects**

*On the Use of Coloured Petri Nets for Object-Oriented Design* [Barros and Gomes, 2004]

-----

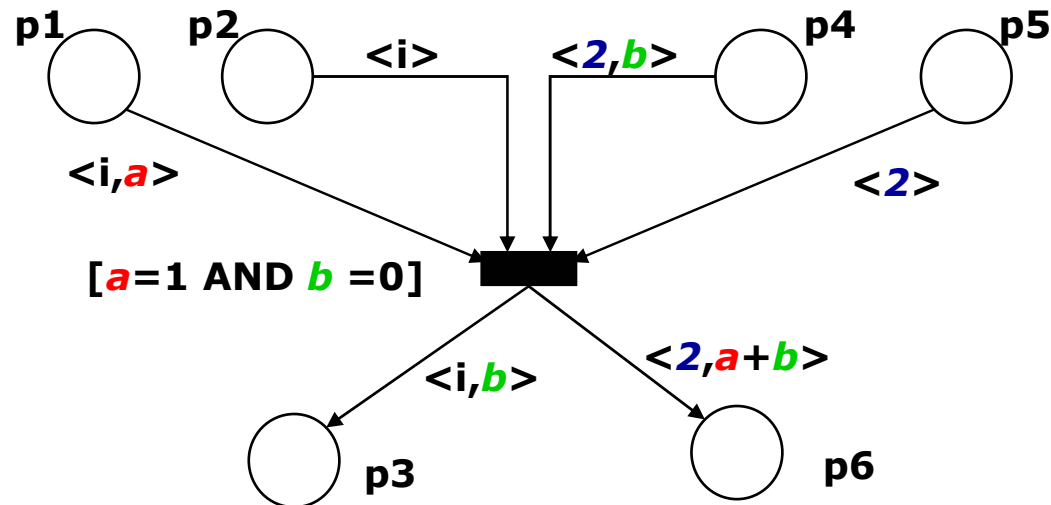
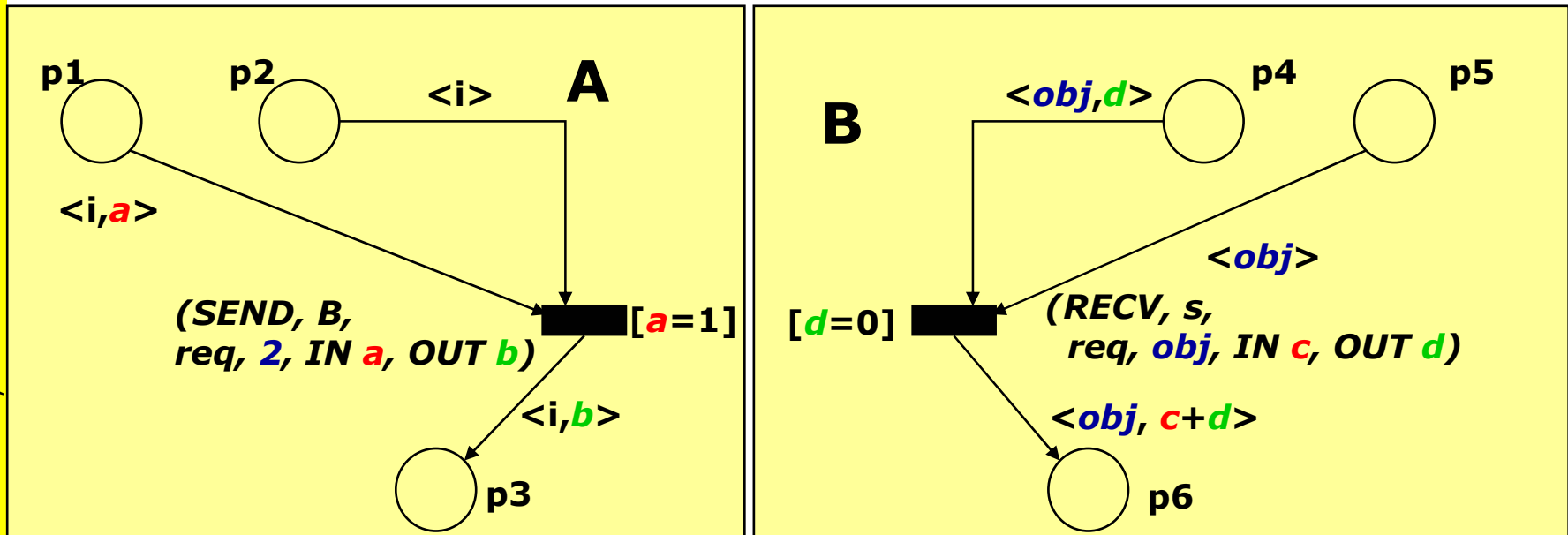
*Coloured Petri Nets with Synchronous Channels*  
[Christensen and Hansen, 1992]

*Modular Coloured Petri nets*  
[Christensen and Petrucci, 1992]

*Object Petri nets*  
[Lakos, 1995]

*Object Coloured Petri Nets*  
[Maier and Moldt, 2001]

# Synchronous requests/ Transition Fusion in a CPN



# Transition Fusion and the CPNTools

1. A model transformation approach;

From SEND and RECV declarations inside comments in code segments to a new model with a new page for each request invocation.

2. The class/page name can be a variable, allowing polymorphic invocations/fusions.

# Request pages

## First tentative solution:

SEND transition as a substitution transition.  
RECV transition is deleted.  
A new transition is created in a new page.

## Final solution:

SEND transition is deleted.  
RECV transition is deleted.  
A new transition is created in a new page.

**Two examples...**

**Instead of transformation**

**integration...**

## **Integration in the CPNTools (1/2)**

1. Same semantics but without model transformations.
2. Each transition should admit an extra inscription (separated from code segments) for the specification of a SEND or RECV request.
3. SEND requests with no corresponding RECV requests should be signalled as syntax errors.
4. The CPNTools left margin should show each receive transition and the respective send transitions. One mouse click should take the modeller to the page (class) containing the respective send or receive transition.

## Integration in the CPNTools (2/2)

5. Non-matching parameter types, or qualifiers, between send and receive transition pairs, should be signalled as syntax errors.
6. The mandatory occurrences of parameters in input arcs should be automatically checked. These bind IN parameters in SEND requests, and OUT parameters in RECV requests.
7. OUT parameters in SEND requests and IN parameters in RECV requests should be considered as possible ways to bind transition variables; (this makes “check places” useless);
8. To allow polymorphic invocations, the use of a variable `targetClass` parameter with class (page) names as domain should be supported

# Concluding...

## Synchrony Groups

Initially proposed in the broader context of object-oriented modelling with CPNs for modelling synchronous *object requests*.

*but...*

offer an general unidirectional transition fusion.

Can be done through model transformation

but

would be better without model transformation!...

# A Unidirectional Transition Fusion for Coloured Petri Nets and its Implementation for the CPNTools

**João Paulo Barros<sup>1,2</sup>**

jpb@uninova.pt

**Luís Gomes<sup>1</sup>**

lugo@uninova.pt

<sup>1</sup>Universidade Nova de Lisboa  
Faculdade de Ciências e Tecnologia  
Dep. de Engenharia Electrónica  
Laboratórios da FCT-UNL  
2825 Monte de Caparica

<sup>2</sup>Instituto Politécnico de Beja  
Escola Superior de Tecnologia e Gestão  
Área Departamental de Engenharia  
Rua Afonso III, n.º 1  
7800-050 Beja

PORTUGAL

**Thanks for your attention!**