d-l@b

Design Lab

PMR 5237
Modelagem e Design de Sistemas Discretos

Aula 9

Prof. Dr. José Reinaldo Silva

reinaldo@usp.br

Escola Politécnica da USP

MECATRÔNICA

Indeed (?)

It is *not necessary* for a *user* to know the formal definition of CP-nets:

- The correct *syntax* is checked by the CPN editor, i.e., the computer tool by which CP-nets are constructed.
- The correct use of the *semantics* (i.e., the enabling rule and the occurrence rule) is guaranteed by the CPN simulator and the CPN tools for formal verification.

Kurt Jensen

It is all Petri Nets

High-level contra low-level nets

The relationship between CP-nets and Place/Transition Nets (PT-nets) is *analogous* to the relationship between high-level programming languages and assembly code.

- In theory, the two levels have exactly the same *computational power*.
- In practice, high-level languages have much more *modelling power* – because they have better structuring facilities, e.g., types and modules.

Os problemas

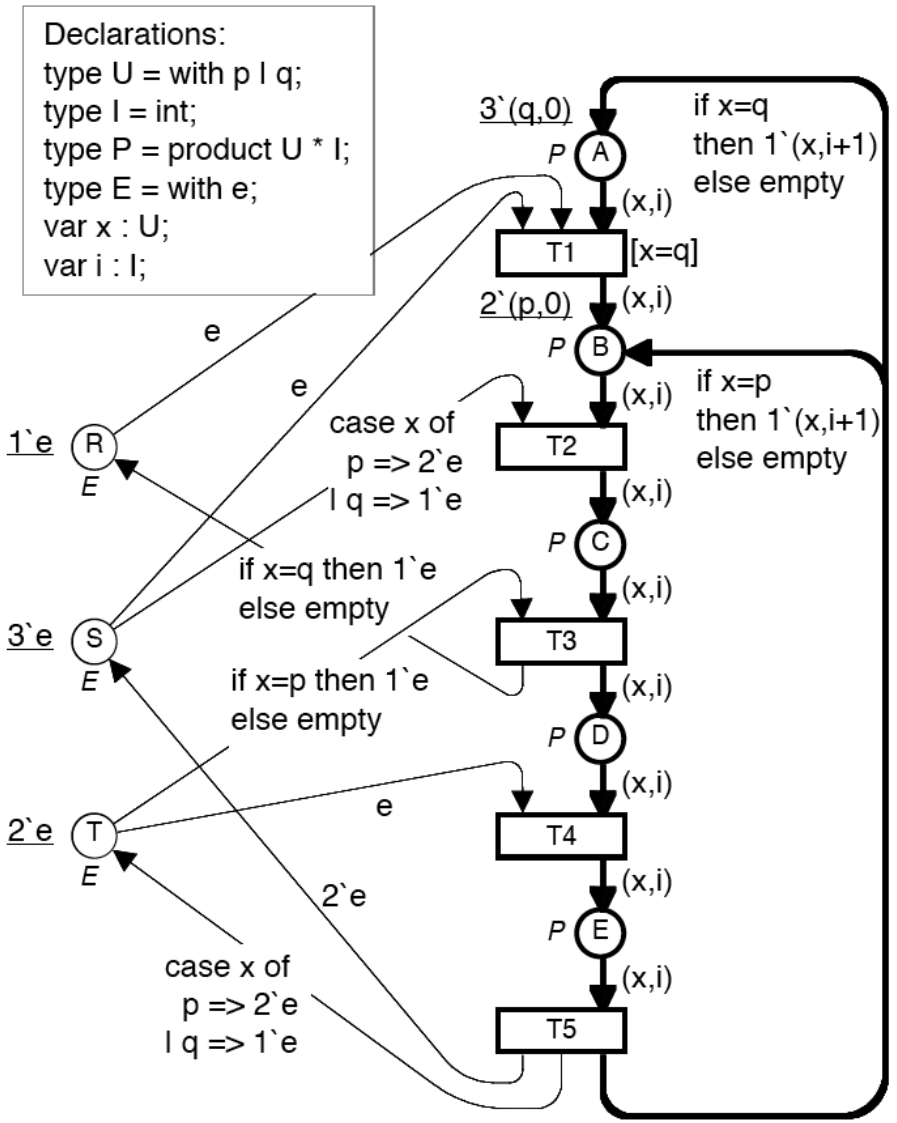
Continuar dependendo muito da simulação e da atingibilidade.

Menor apelo direto à intuição.

Uso de outro tipo de estruturação

Falta de um bom arcabouço de análise de propriedades

The Resource Allocation Problem



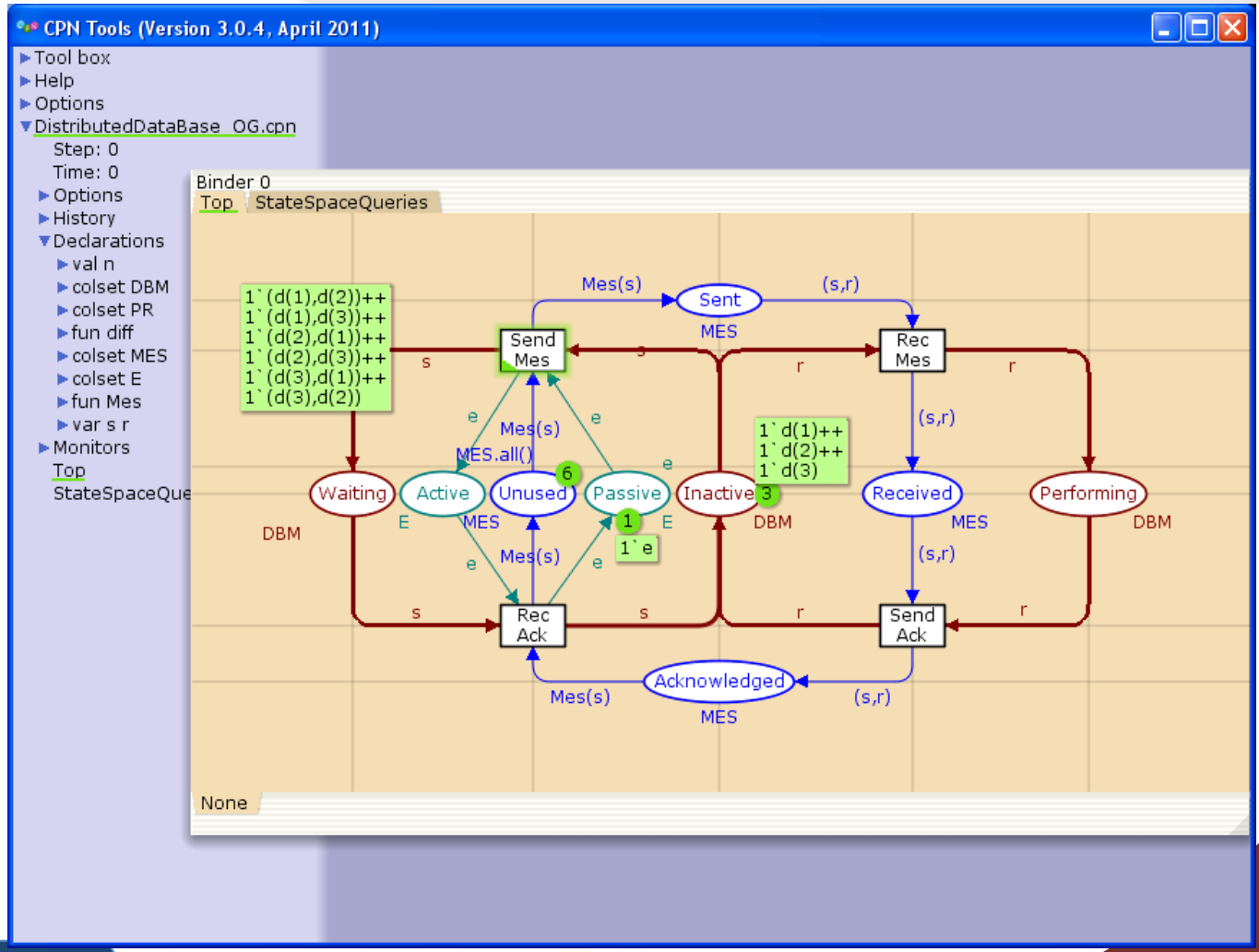
Analyzing a CPN

CP-nets may be large

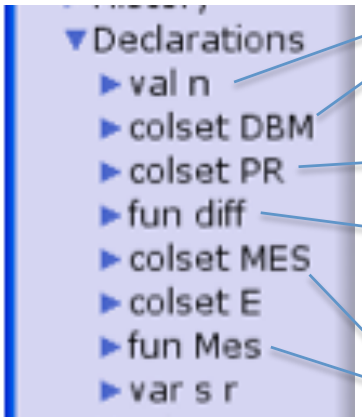
A typical *industrial application* of CP-nets contains:

- 10-200 *pages*.
- 50-1000 *places and transitions*.
- 10-200 *colour sets*.

CPN Analysis



Variables and Colorsets



The colorset DBM is the set of DB managers $\{d_i\}$ i=1..n.

The variable n denotes the number of database managers.

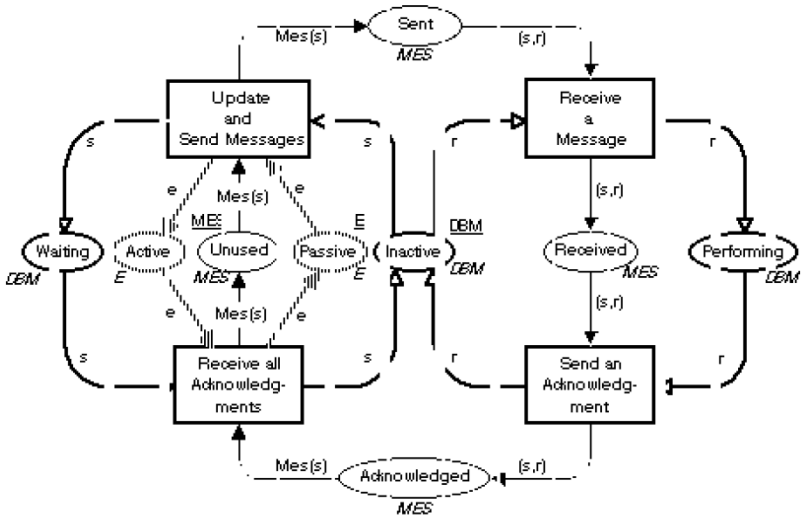
The colorset PR: DBM X DBM, and denotes all possible communication channel between two DB managers.

The function diff: (DBM X DMB) ➔ {V, F}, is a boolean function that interpret a pair (d_i, d_j) as V if $d_i \neq d_j$.

The function Mes : DBM ➔ MES_{MS} is such that $MES(d_i)=\{1'(d_i, d_j), j=1..n|diff(d_i, d_j)$

The colorset MES is a subset of PR such that $MES(d_i)=\{(d_i, d_j)|diff(d_i, d_j)$

The initial state



$M_0(\text{Inactive}) = \text{DBM}$

$M_0(\text{Performing}) = M_0(\text{Waiting}) = \emptyset$

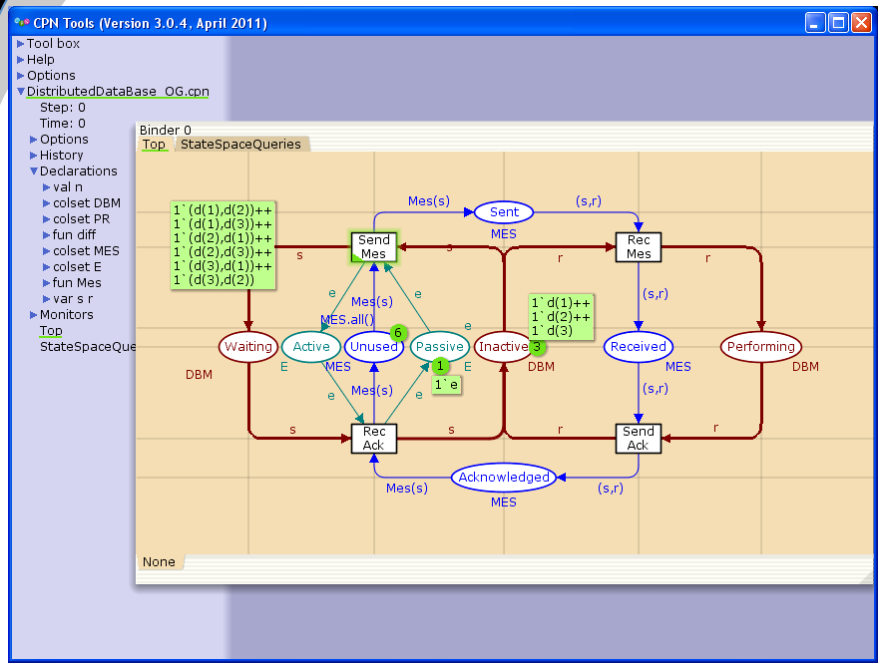
$M_0(\text{Unused}) = \text{MES} = \{(s,r) \in \text{DBM} \times \text{DBM} \mid s \neq r\}$

$M_0(\text{Sent}) = M_0(\text{Received}) = M_0(\text{Acknowledged}) = \emptyset$

$M_0(\text{Passive}) = \text{E} = 1`e$

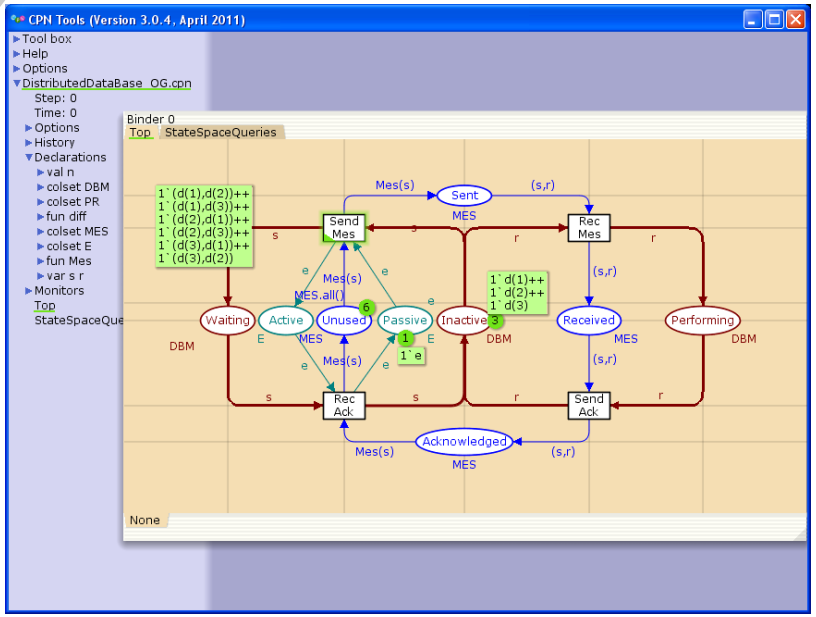
$M_0(\text{Active}) = \emptyset.$

Conflict



In the initial state all managers are in the place Inactive, there is a black resource in the place Passive (meaning that the action is allowed to only one manager) and all messages are in Unused. Thus, it is possible to fire Send Mes for any manager d_i, $1 \leq i \leq n$, for a fixed value of n.

Enabling condition



In the state M0 only the transition Send Mes (SM) is enabled, considering all the arc expressions. Thus, if a binding is selected from the conflicting situation, for instance,

$$(SM, <s=d_2>)$$

SM will occur, putting marks in Sent, Active and Waiting.

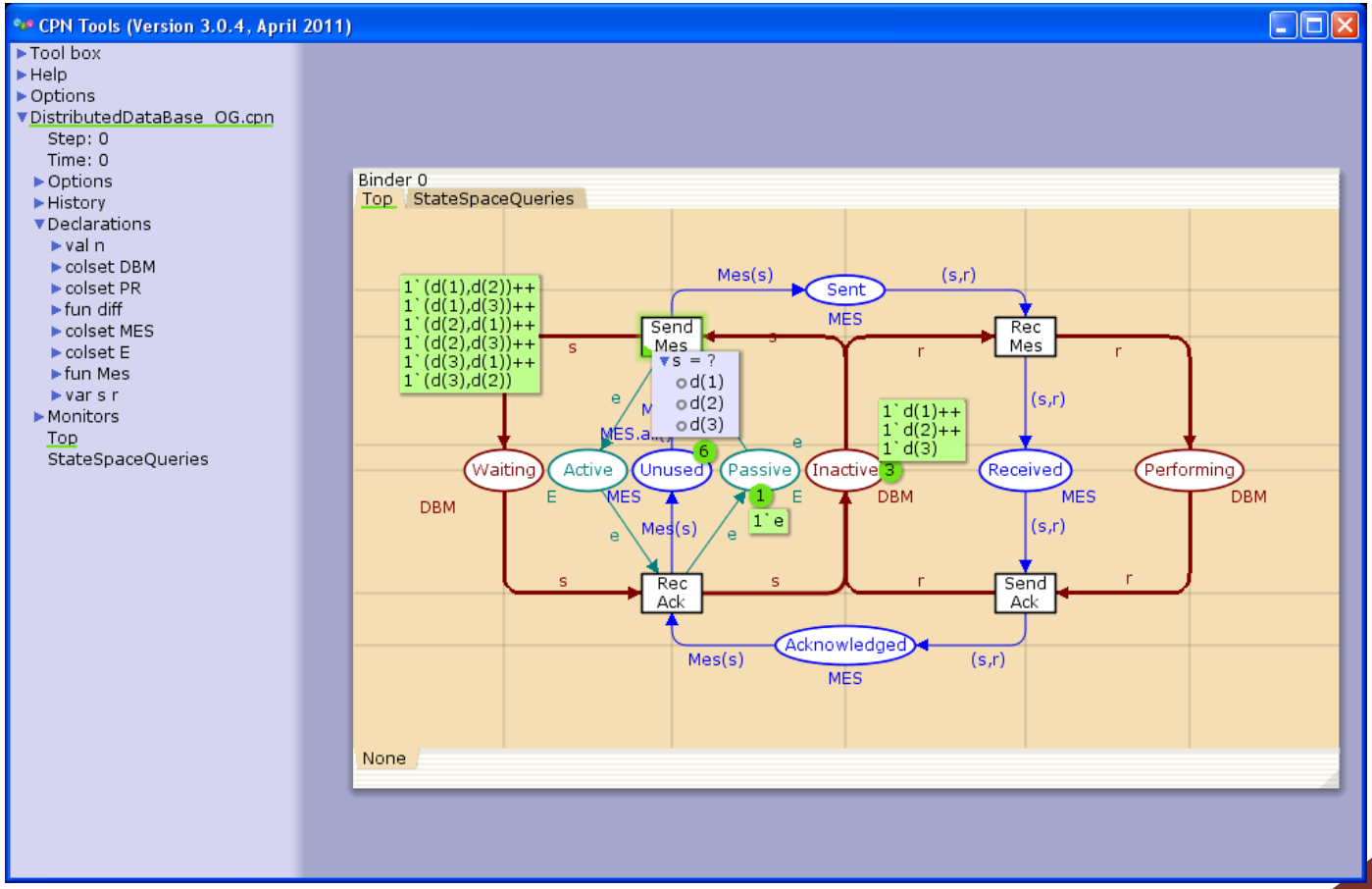
Binding definition

Definition 3.4: A **binding** of a transition t is a function b defined on Var(t), such that:

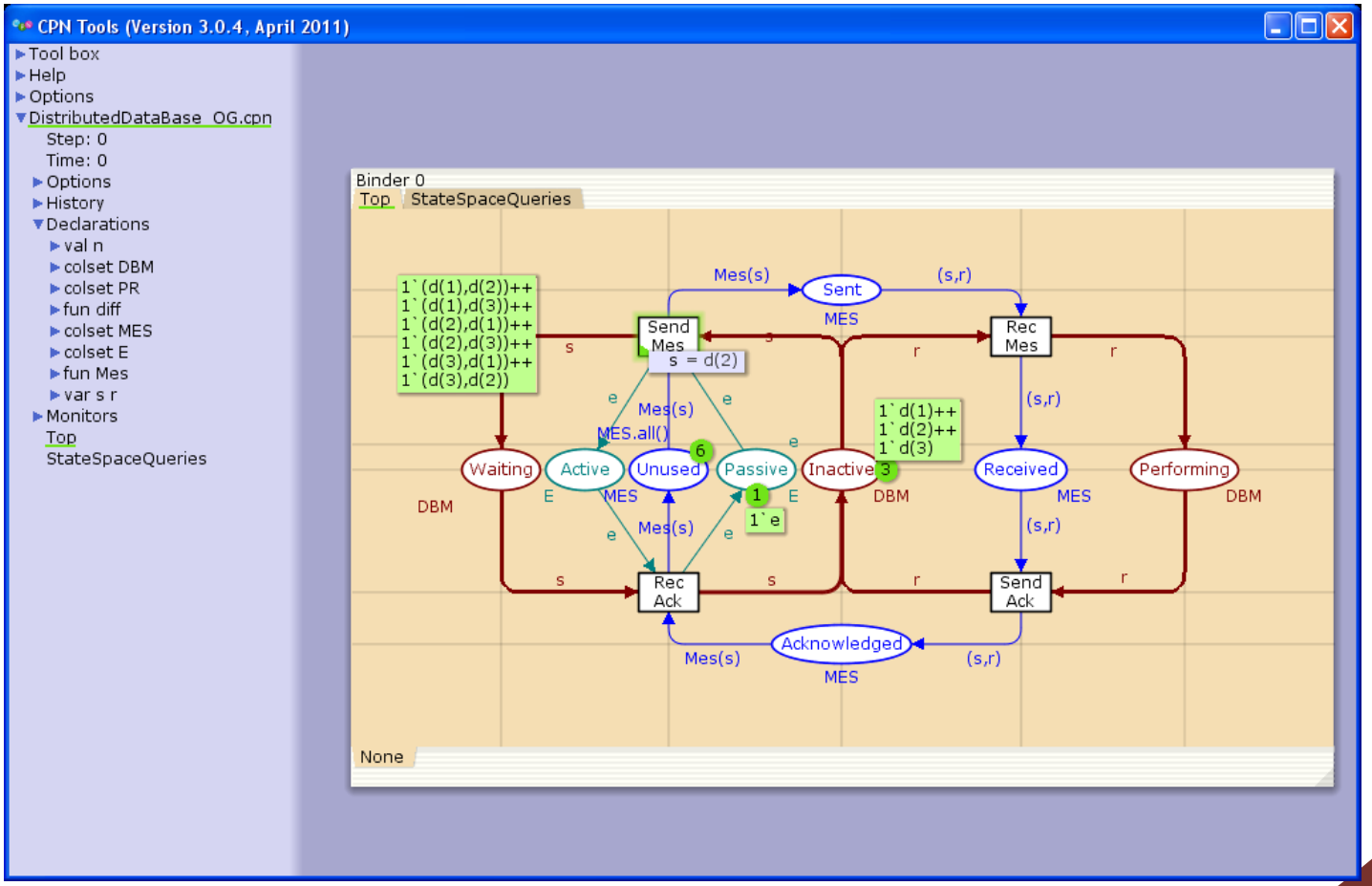
(i) $\forall v \in Var(t)$: $b(v) \in Type(v)$.
(ii) $G(t)\langle b \rangle$.

By B(t) we denote the set of all bindings for t.

The conflict in CPNTools



Setting the biding

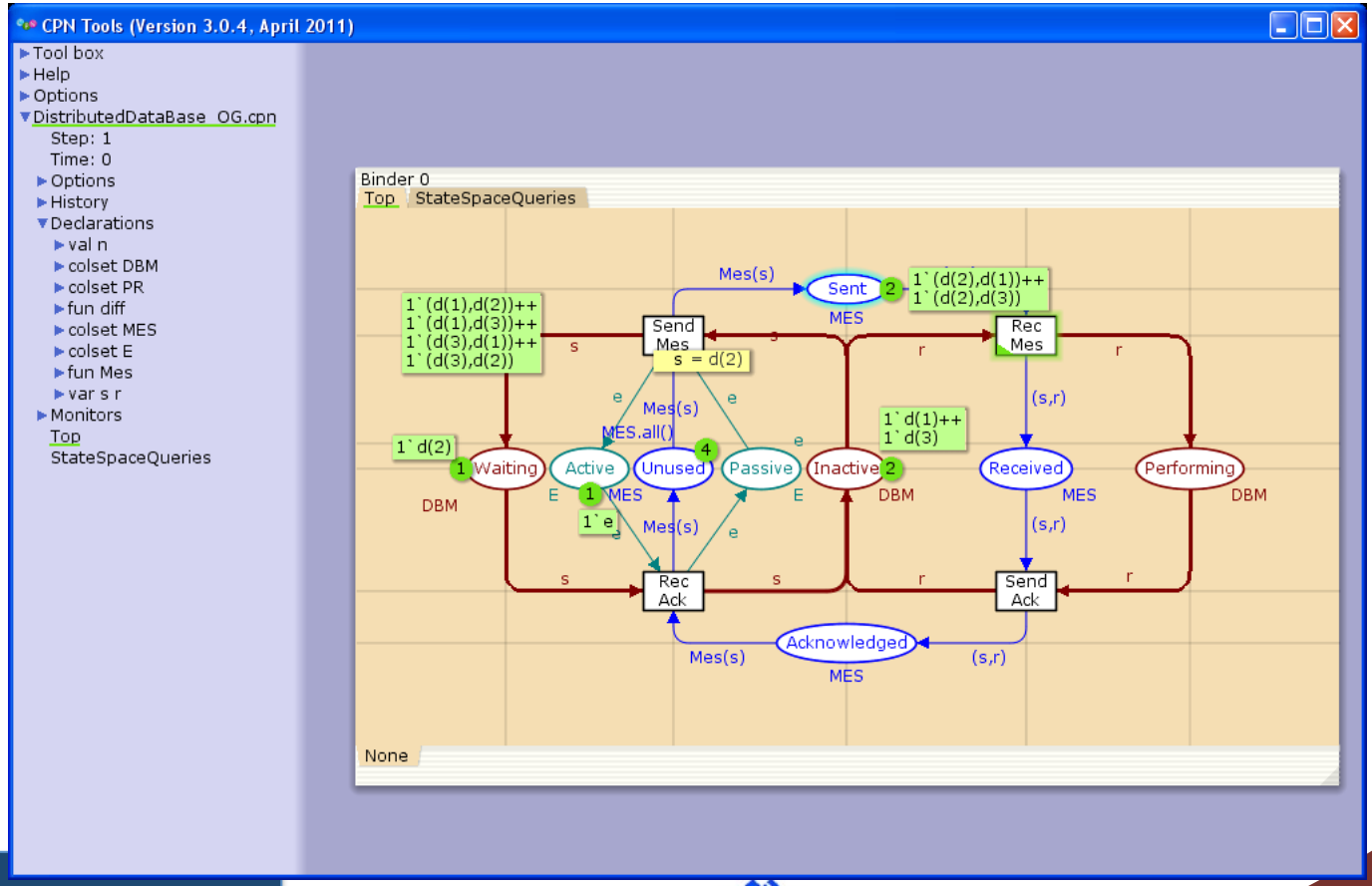


Escola Politécnica da USP

MECATRÔNICA

Firing SM

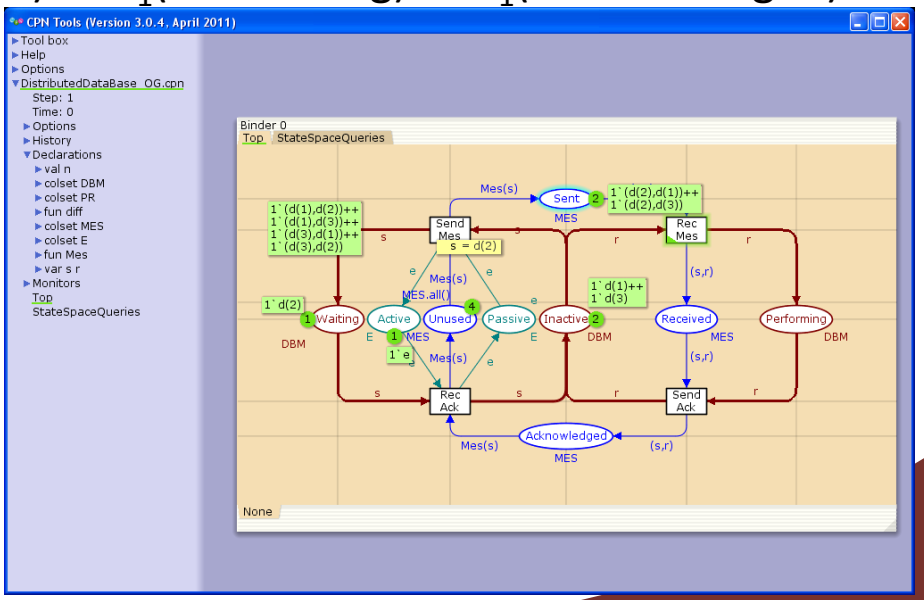
Firing SM lead to a new state M_1



The state M_1

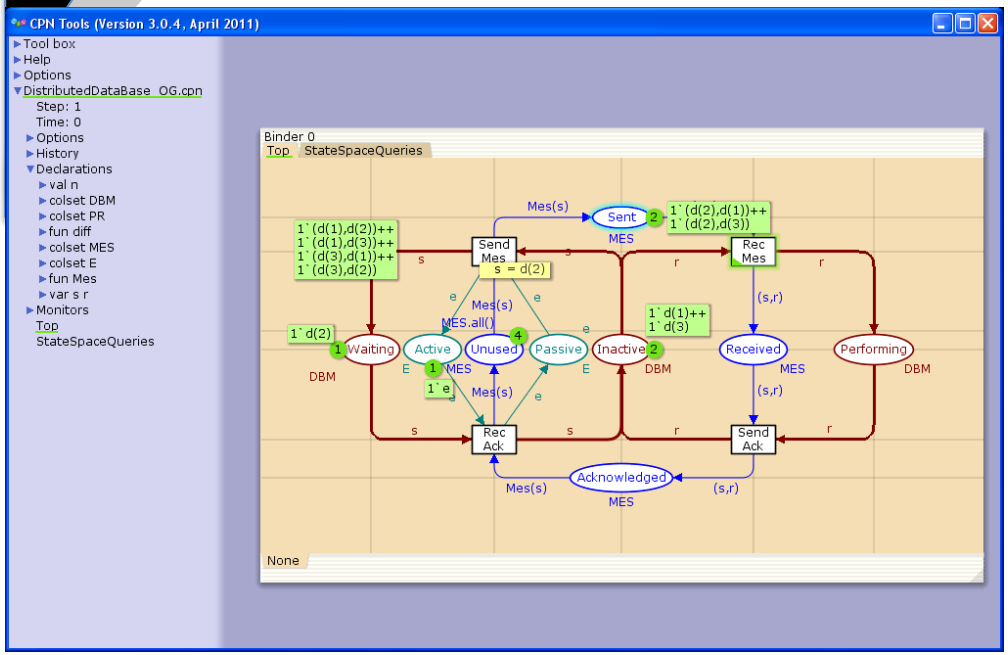
The state M_1 can be obtained from M_0 by following the flux of marking since it respects the arc expressions, the biding, and the filters.

$M_1(Inactive) = M_0(Inactive) - 1`d_2 = DBM - 1`d_2$
$M_1(Waiting) = 1`d_2$
$M_1(Sent) = Mes(d_2) = 1`(d_2, d_1) + 1`(d_2, d_3)$
$M_1(Unused) = M_0(Unused) - Mes(d_2) = MES - Mes(d_2)$
$M_1(Active) = 1`e$
$M_1(Passive) = M_1(Received) = M_1(Performing) = M_1(Acknowledged) = \Phi$



Escola Politécnica da USP

The step



In the new state M_1 there is only one transition enabled which is Rec Mes (RM). That means that the message sent by d_2 could be received by either managers d_2 or d_2. The firing of RM for each manager is what is called concurrent events and make a step denoted by the biding,

(RM, < s= d_2, r=d_1> and/or < s= d_2, r=d_3>)

For the selected example where n=3.

Formal definition of step

Definition 3.6: A step Y is **enabled** in a marking M iff the following property is satisfied:

$$\forall p \in P: \sum_{(t,b) \in Y} E(p,t)\langle b \rangle \leq M(p).$$

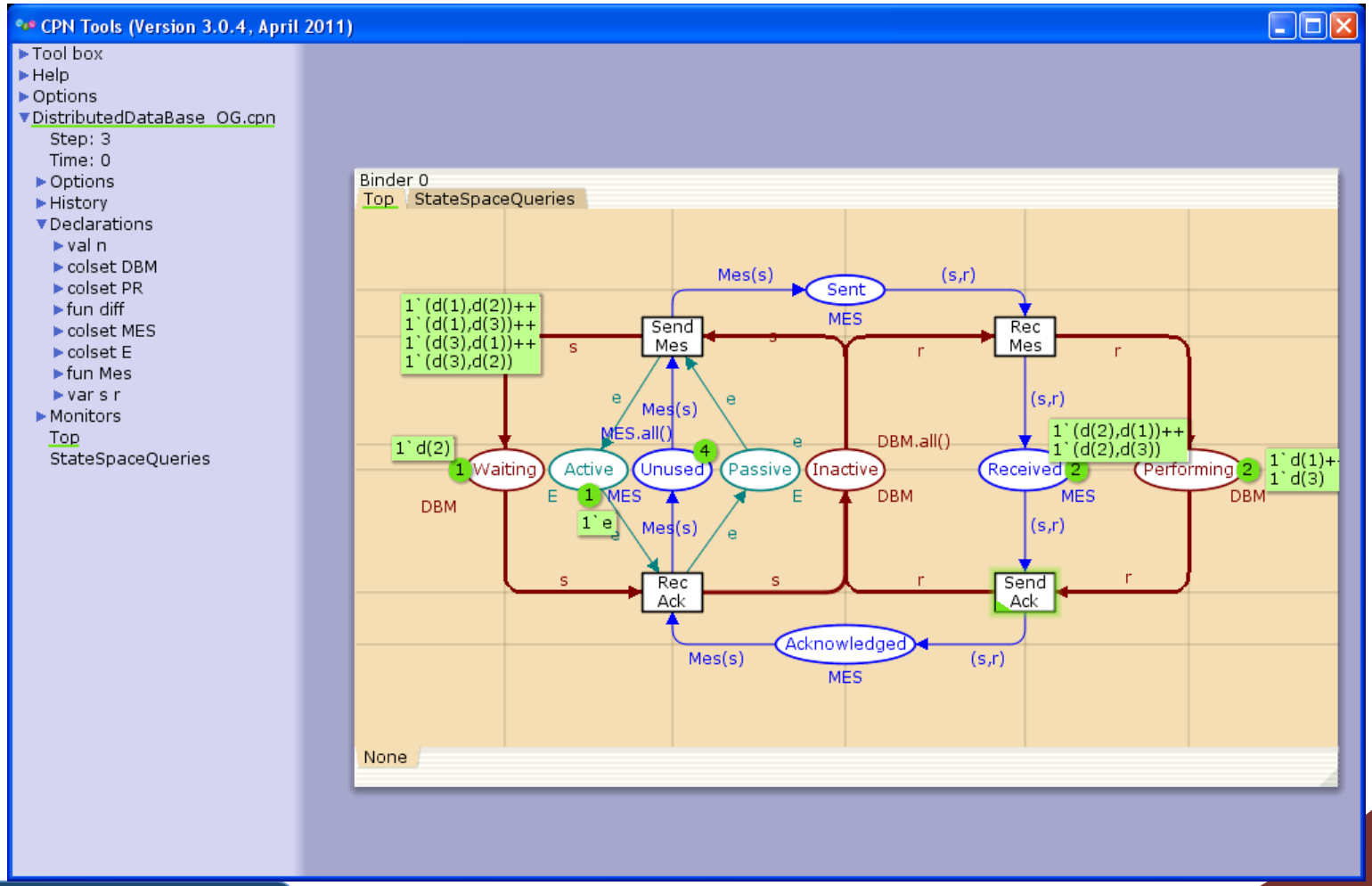
We then say that (t,b) is enabled and we also say that t is enabled. The elements of Y are concurrently enabled (when $|Y| \geq 1$).

When a step Y is enabled in a marking M_1 it may **occur**, changing the marking M_1 to another marking M_2, defined by:

$$\forall p \in P: M_2(p) = \left(M_1(p) - \sum_{(t,b) \in Y} E(p,t)\langle b \rangle\right) + \sum_{(t,b) \in Y} E(t,p)\langle b \rangle.$$

M_2 is **directly reachable** from M_1. This is written: $M_1 [Y\rangle M_2$.

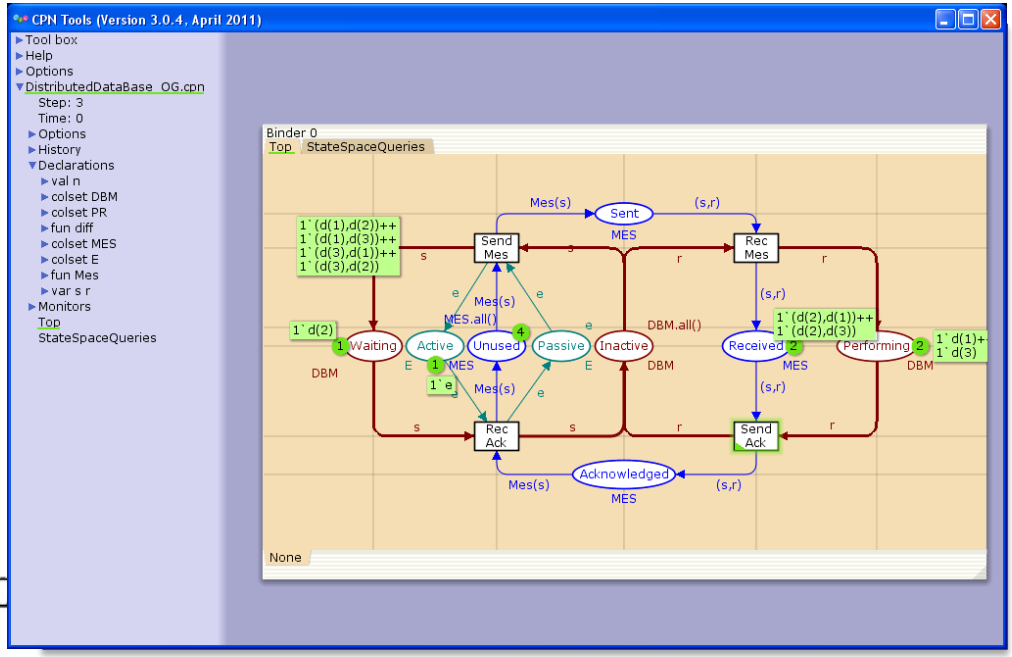
Firing the step



Obtaining the new state

The state M_2 can be obtained from M_1 by following the flux of marking since it respects the arc expressions, the biding, and the filters.

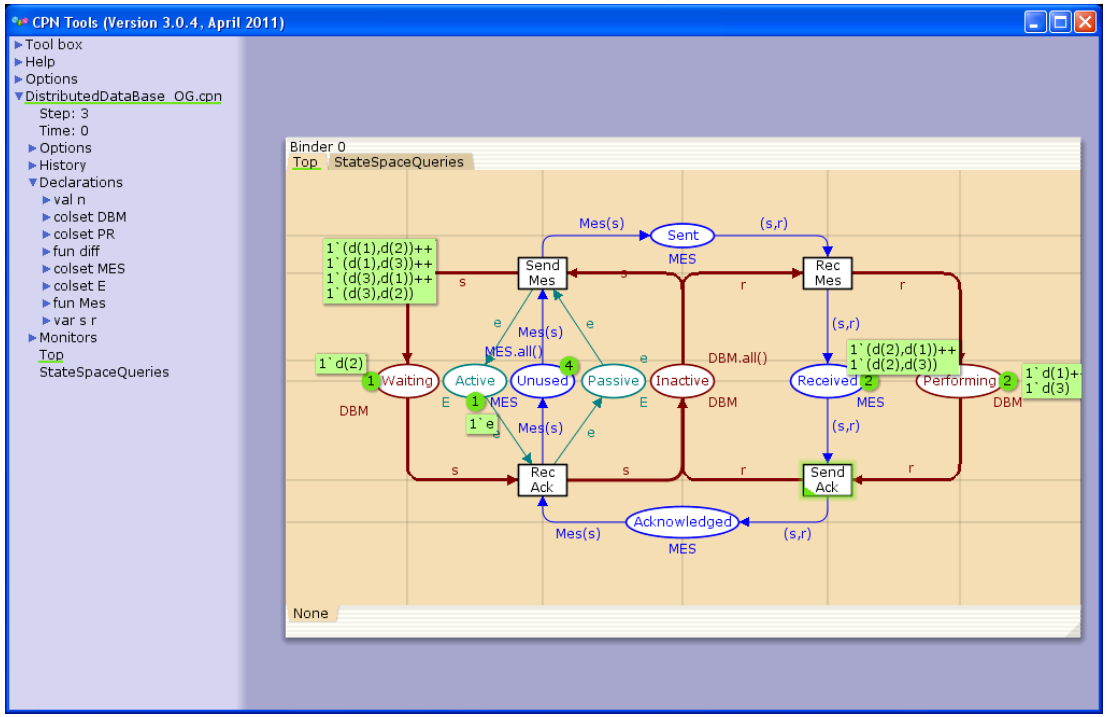
$M_2(\text{Inactive}) = M_1(\text{Inactive}) = M_0(\text{Inactive}) - 1`d_2 = \text{DBM} - 1`d_2$
$M_2(\text{Waiting}) = M_1(\text{Waiting}) = 1`d_2$
$M_2(\text{Sent}) = \Phi$
$M_2(\text{Unused}) = M_1(\text{Unused}) = M_0(\text{Unused}) - \text{Mes}(d_2) = \text{MES} - \text{Mes}(d_2)$
$M_2(\text{Active}) = M_1(\text{Active}) = 1`e$
$M_2(\text{Received}) = 1`(d_2, d_1) + 1`(d_2, d_3)$
$M_2(\text{Performing}) = 1d_2 + 1`d_3$
$M_2(\text{Passive}) = M_2(\text{Acknowledged}) = \Phi$



Escola Politécnica da USP

Concurrency

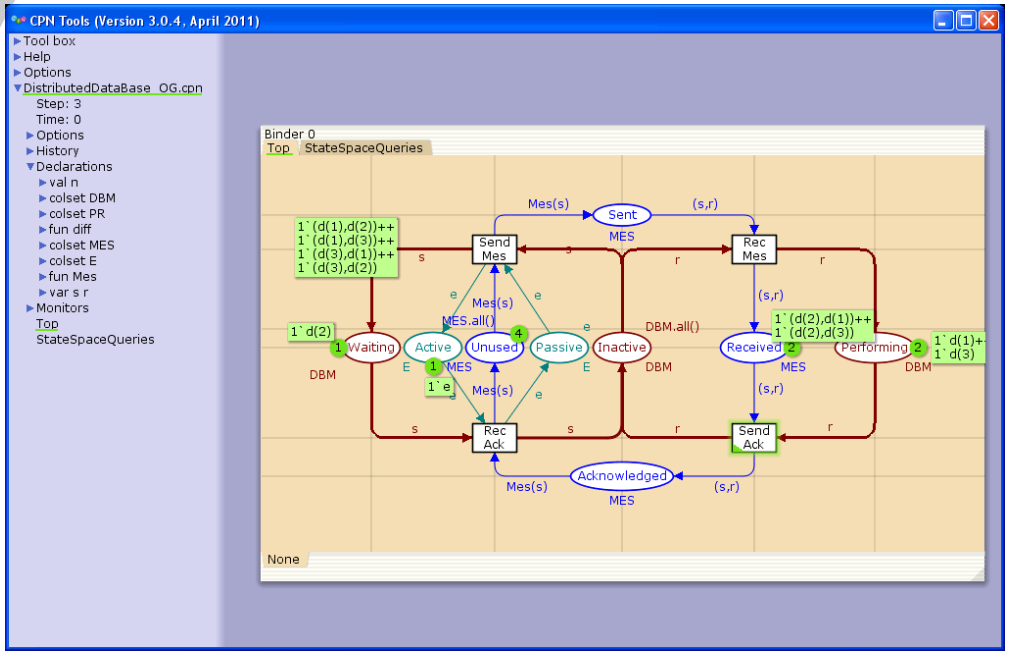
The same situation of concurrency happens again in state M2 since either manager d_1 or d_3 (or both) can choose to send an acknowledge back to d_2.



Escola Politécnica da USP

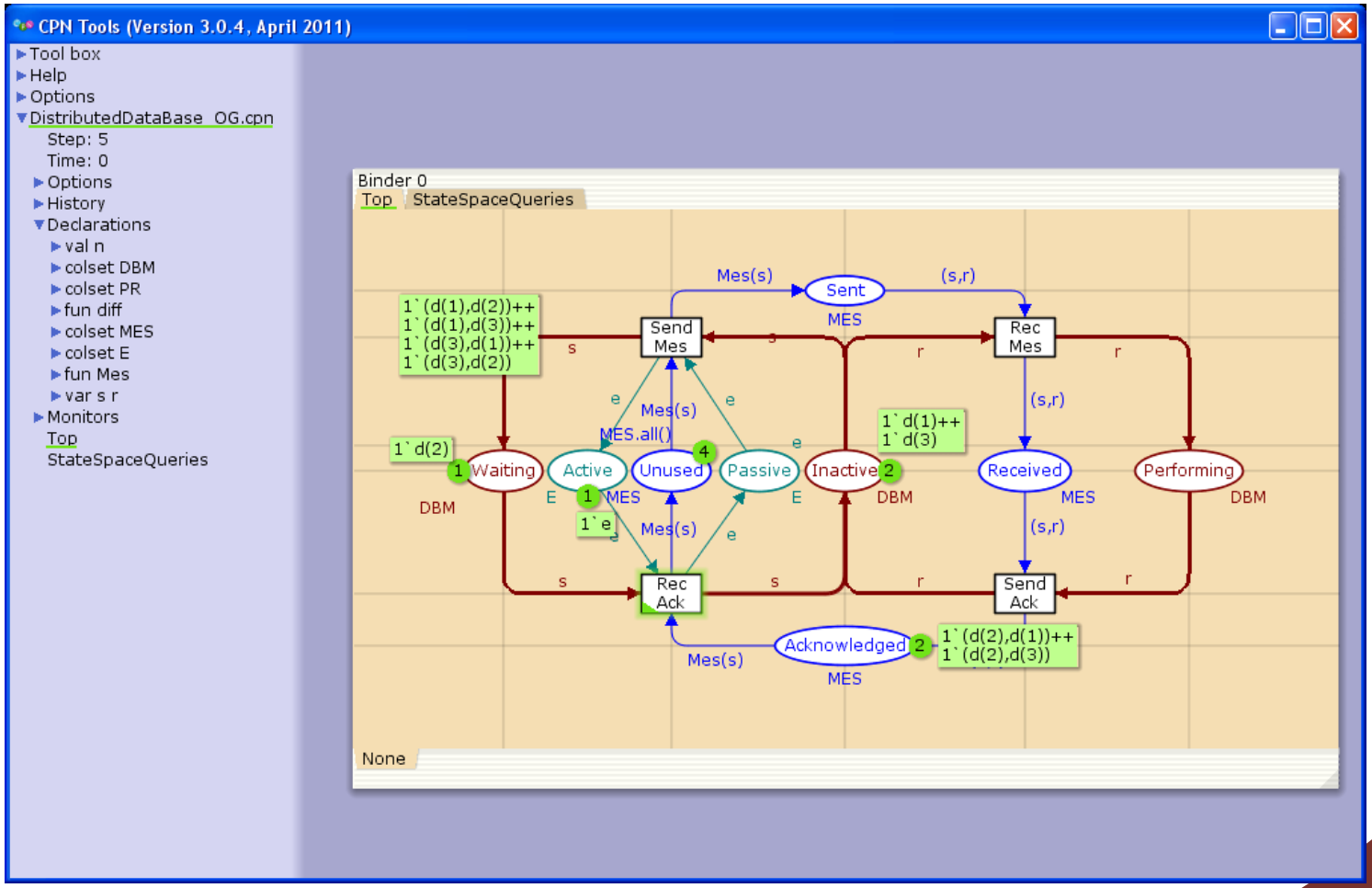
MECATRÔNICA

The new step



Only one transition, the Send Ack (SA) enabled and could be fired by one of the managers, d_1 or d_3 once they finish upgrading the changes made by d_2. Again the sending of an acknowledge is independent for each one of the managers.

The new state



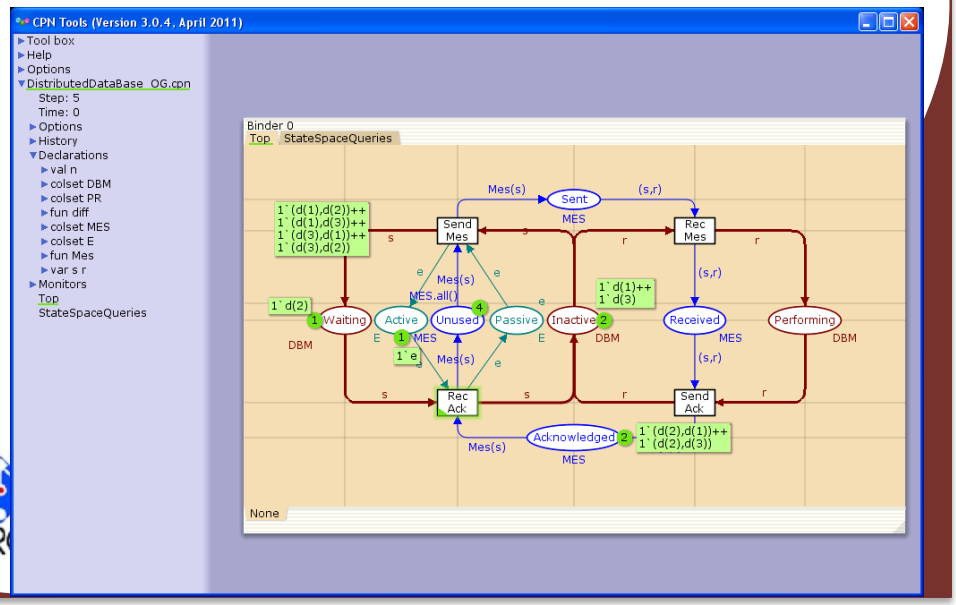
Escola Politécnica da USP

MECATRÔNICA

The new state

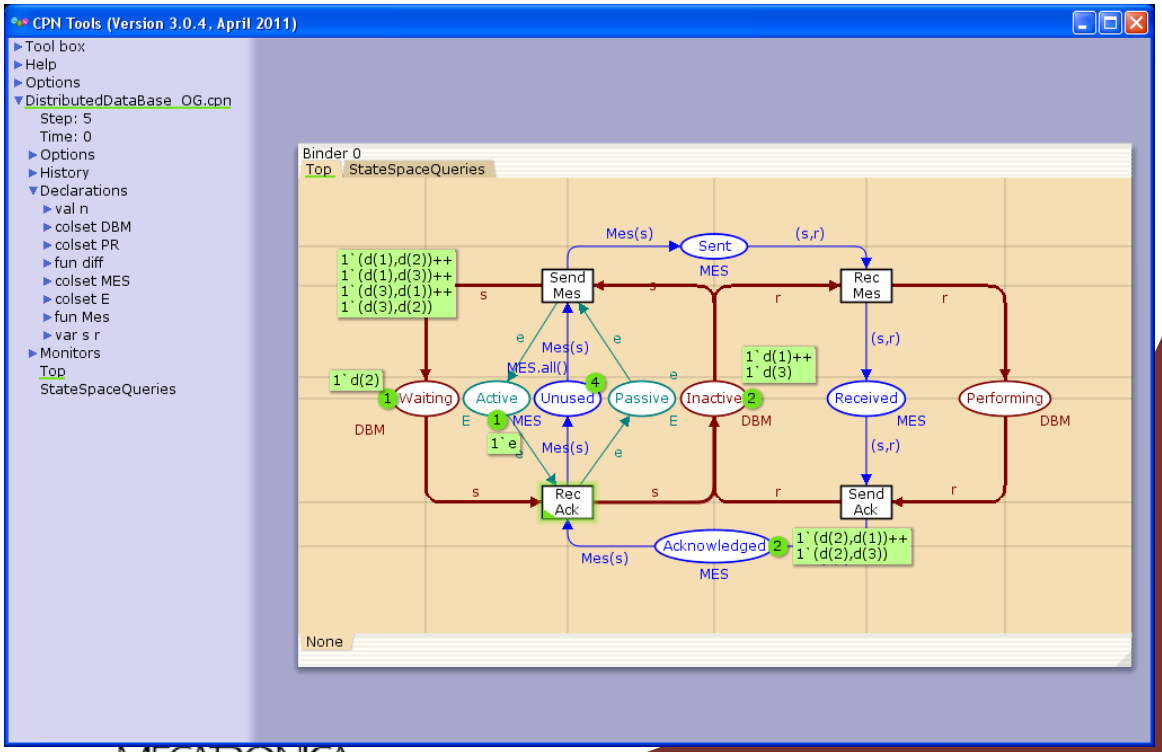
The state M_3 can be obtained from M_2 by following the flux of marking since it respects the arc expressions, the biding, and the filters.

$M_3(Inactive) = M_2(Inactive)$
$M_3(Waiting) = M_2(Waiting)$
$M_3(Sent) = \Phi$
$M_3(Unused) = M_2(Unused)$
$M_3(Active) = M_2(Active) = 1`e$
$M_3(Received) = \Phi$
$M_3(Performing) = \Phi$
$M_3(Passive) = \Phi$
$M_3(Acknowledged) = 1`(d_2, d_1) + 1`(d_2, d_3)$



Synchronization

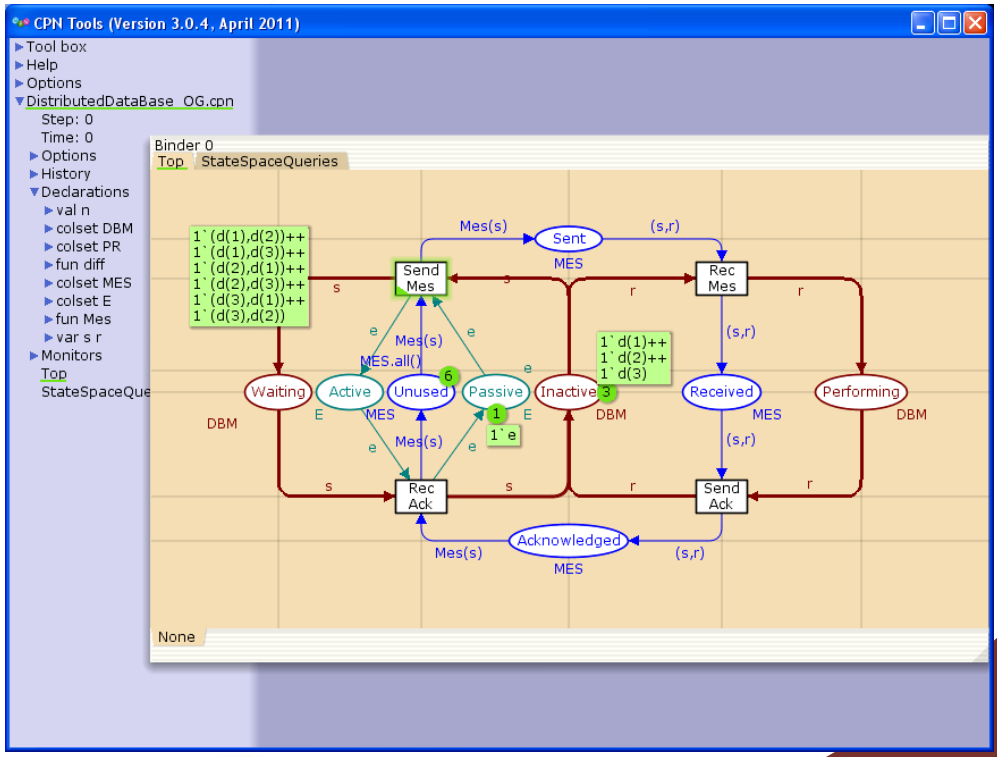
In state M3 there is only one transition enabled which is Receive Ack (RA). In this case the acknowledge of all associated managers must be combined in a message to the sender. After receiving this message the sender comes back to a passive state again.



The new state

The state M_4 can be obtained from M_3 by following the flux of marking since it respects the arc expressions, the biding, and the filters.

$M_4(Inactive) = 1`d_1 + 1`d_2 + 1`d_3 = DBM$
$M_4(Waiting) = \Phi$
$M_4(Sent) = \Phi$
$M_4(Unused) = MES$
$M_4(Active) = \Phi$
$M_4(Received) = \Phi$
$M_4(Performing) = \Phi$
$M_4(Passive) = 1`e$
$M_4(Acknowledged) = \Phi$



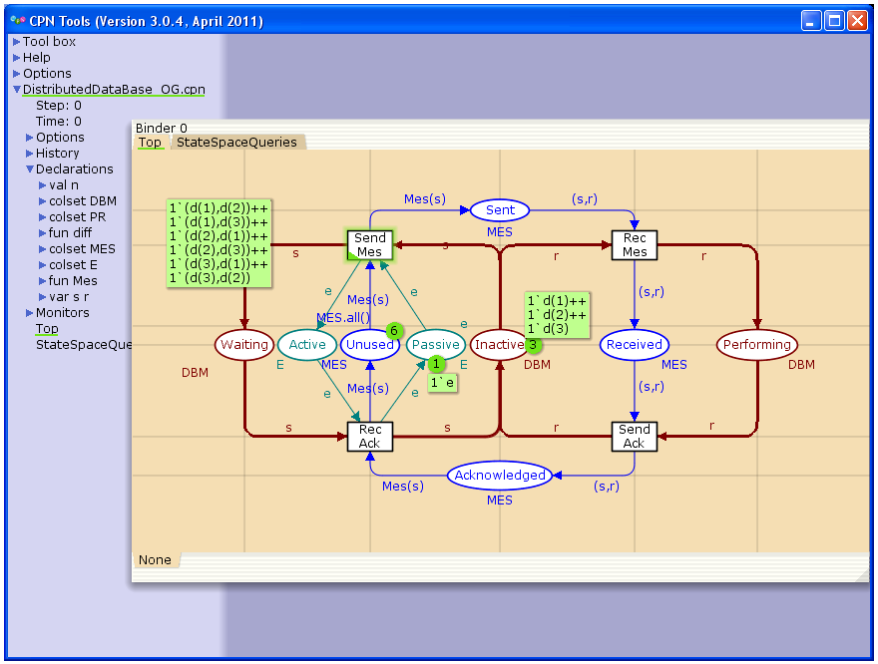
A cycle

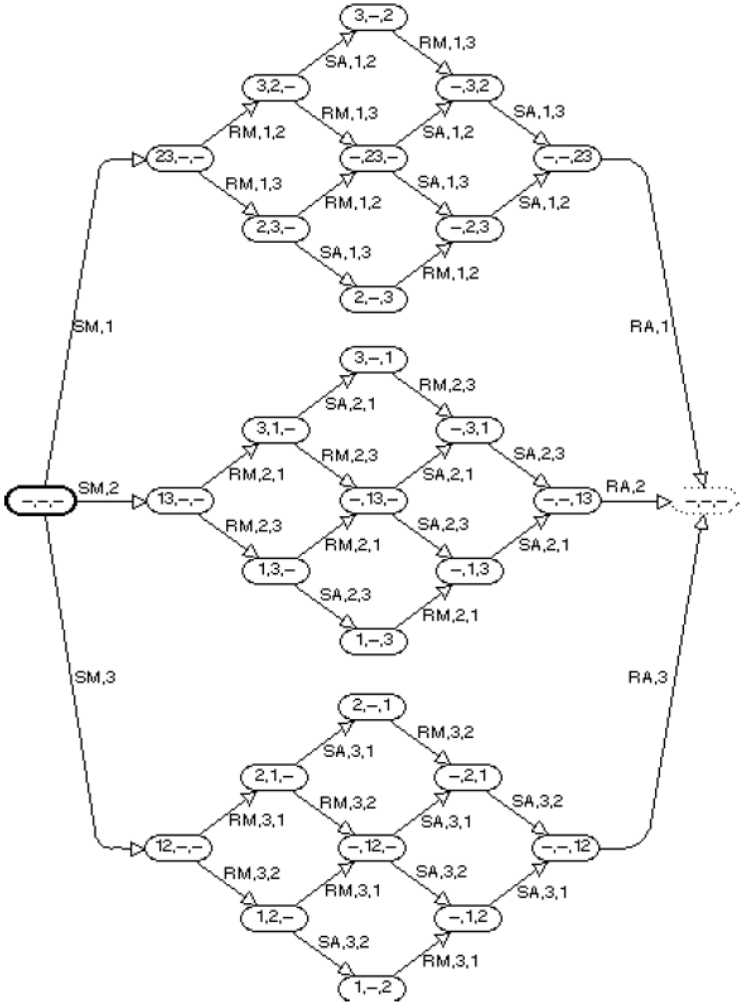
It could be seen that starting in a situation where all managers are inactive, if one of them makes a change and then send a message the final answer is the that the remaining managers receive the message, perform the change and send an acknowledge closing a cycle, that is, changing to the initial state. Thus,

M0 |SM> M1 |RM> M2 |SA> M3 |RA> M0

That will be symmetric for each manager and constitutes a transition invariant.

Occurrence Graph





Escola Politécnica da USP

MECATRÔNICA

Directed Graphs

Definition 6.1: A **directed graph** is a tuple DG = (V, A, N) such that:

(i) V is a set of **nodes** (or **vertices**).

(ii) A is a set of **arcs** (or **edges**) such that:

- $V \cap A = \emptyset$.

(iii) N is a **node** function. It is defined from A into $V \times V$.

DG is **finite** iff V and A are finite.

O-graph

Definition 6.2: The **full occurrence graph** of a CP-net, also called the **O-graph**, is the directed graph OG = (V, A, N) where:

(i) $V = [M_0\rangle$.

(ii) $A = \{(M_1,b,M_2) \in V \times BE \times V \mid M_1\,[b\rangle\, M_2\}$.

(iii) $\forall a=(M_1,b,M_2) \in A: N(a) = (M_1,M_2)$.

Generating an O-graph

Proposition 6.3: The following algorithm constructs the O-graph. The algorithm halts iff the O-graph is finite. Otherwise the algorithm continues forever, producing a larger and larger subgraph of the O-graph.

```
W := Ø
Node(M0)
repeat
    select a node M1∈W
    for all (b,M2)∈Next(M1) do
    begin
        Node(M2)
        Arc(M1,b,M2)
    end
    remove M1 from W
until W = Ø.
```

Behavioral Properties

As pointed before, we could treat CPN as classical net in what concerns the design and modeling process. Therefore, we should reinforce property analysis as a method instead of trusting only in simulation and reachability.

All main properties analyzed before in classic nets could also apply to CPN.

Boundness

Boundedness properties tell us how many tokens we may have at a particular place:

Definition 4.1: Let a place $p \in P$, a non-negative integer $n \in \mathbb{N}$ and a multi-set $m \in C(p)_{MS}$ be given.

(i) n is an **integer bound** for p iff:
$\forall M \in [M_0\rangle: \ |M(p)| \leq n.$

(ii) m is a **multi-set bound** for p iff:
$\forall M \in [M_0\rangle: M(p) \leq m.$

Applying boundness

	Multi-set	Integer
Inactive	DBM	n
Waiting	DBM	1
Performing	DBM	n−1
Unused	MES	n^2-n
Sent, Received, Acknowledged	MES	n−1
Passive, Active	E	1

Home Space

Definition 4.2: Let a marking $M \in \mathbb{M}$ and a set of markings $X \subseteq \mathbb{M}$ be given:

(i) M is a **home marking** iff:
$\forall M' \in [M_0\rangle : M \in [M'\rangle$.

(ii) X is a **home space** iff:
$\forall M' \in [M_0\rangle : X \cap [M'\rangle \neq \emptyset$.

Liveness

Definition 4.3: Let a marking $M \in \mathbb{M}$ and a set of binding elements $X \subseteq BE$ be given.

(i) M is **dead** iff no binding element is enabled, i.e., iff:
$\forall x \in BE: \neg M[x\rangle$.

(ii) X is **dead** in M iff no element of X can become enabled, i.e., iff:
$\forall M' \in [M\rangle \ \forall x \in X: \neg M'[x\rangle$.

(iii) X is **live** iff there is no reachable marking in which X is dead, i.e., iff:
$\forall M' \in [M_0\rangle \ \exists M'' \in [M'\rangle \ \exists x \in X: M''[x\rangle$.

Fairness

Definition 4.4: Let $X \subseteq BE$ be a set of binding elements and σ be an infinite occurrence sequence.

(i) X is **impartial** for σ iff it has infinitely many occurrences, i.e., iff:
$OC_X(\sigma) = \infty$.

(ii) X is **fair** for σ iff an infinite number of enablings implies an infinite number of occurrences, i.e., iff:
$EN_X(\sigma) = \infty \quad \Rightarrow \quad OC_X(\sigma) = \infty$.

(iii) X is **just** for σ iff a persistent enabling implies an occurrence, i.e., iff:
$\forall i \geq 1: [EN_{X,i}(\sigma) \neq 0 \quad \Rightarrow \quad \exists k \geq i: [EN_{X,k}(\sigma) = 0 \ \vee \ OC_{X,k}(\sigma) \neq 0]]$.

When X is impartial for all infinite occurrence sequences of the given CP-net (starting in a reachable marking), we say that X is impartial. Analogous definitions are made for fair and just.

The method

We first *construct* a set of place invariants.

Then we check whether they are *fulfilled*.

- This is done by showing that each occurring binding element *respects* the invariants.
- The *removed* set of tokens must be identical to the *added* set of tokens – when the weights are taken into account.

Finally, we use the place invariants to *prove* behavioural properties of the CP-net.

- This is done by a *mathematical proof*.

Automating Invariant Analysis

Automatic calculation of all place invariants:

- This is possible, but it is a very *complex* task.
- Moreover, it is difficult to represent the results on a *useful form*, i.e., a form which can be used by the system designer.

Interactive calculation of place invaritans:

- The *user* proposes some of the weights.
- The *tool* calculates the *remaining weights* – if possible.

Interactive calculation of place invariants is *much easier* than a fully automatic calculation.

Invariant Discussion in CPN

- The user needs some ingenuity to *construct* invariants. This can be supported by *computer tools* – interactive process.
- The user also needs some ingenuity to *use* invariants. This can also be supported by *computer tools* – interactive process.
- Invariants can be used to verify a system – without fixing the *system parameters* (such as the number of sites in the data base system).

Typical Problem Size

CP-nets may be large

A typical *industrial application* of CP-nets contains:

- 10-200 *pages*.
- 50-1000 *places and transitions*.
- 10-200 *colour sets*.

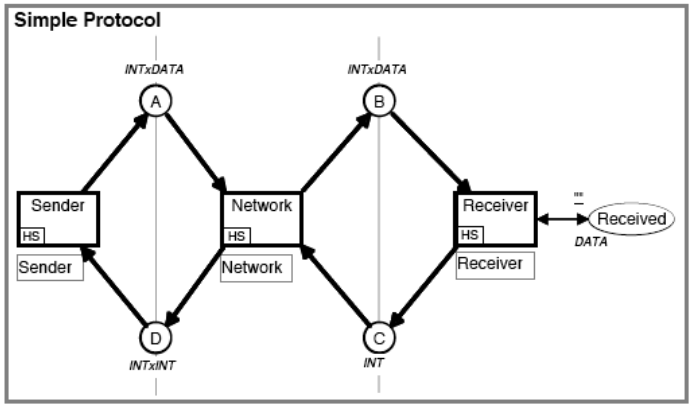
This corresponds to *thousands/millions of nodes* in a Place/Transition Net.

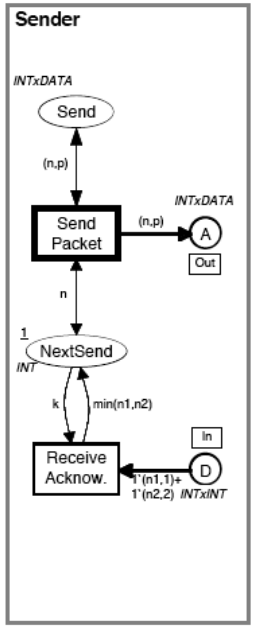
Most of the industrial applications would be *totally impossible* without:

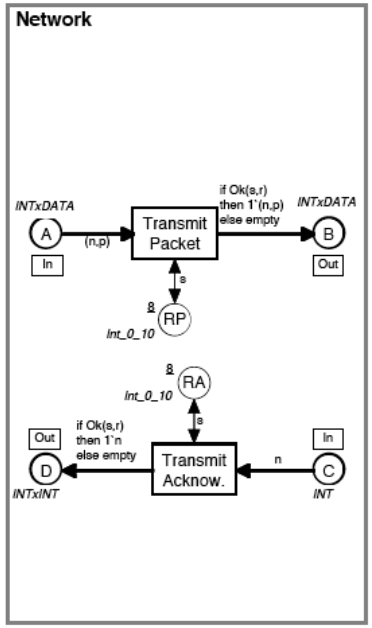
- Colours.
- Hierarchies.
- Computer tools.

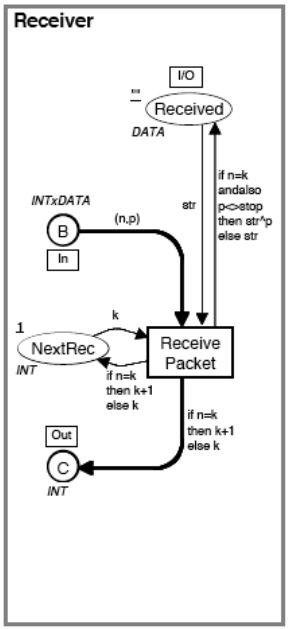
Hierarchical CPNs

A hierarchical CP-net contains a number of *interrelated subnets*– called *pages*.









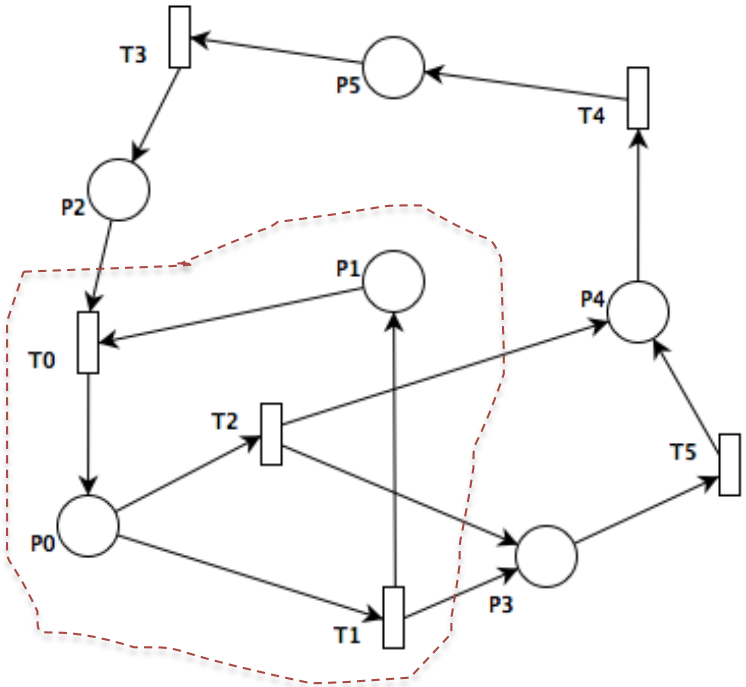
Hierarchy in PN

Hierarchy is not anything new and is actually connected with any kind of net, including the classical ones.

In design, hierarchy means to abstract the elements which properties are not relevant in an analysis phases.

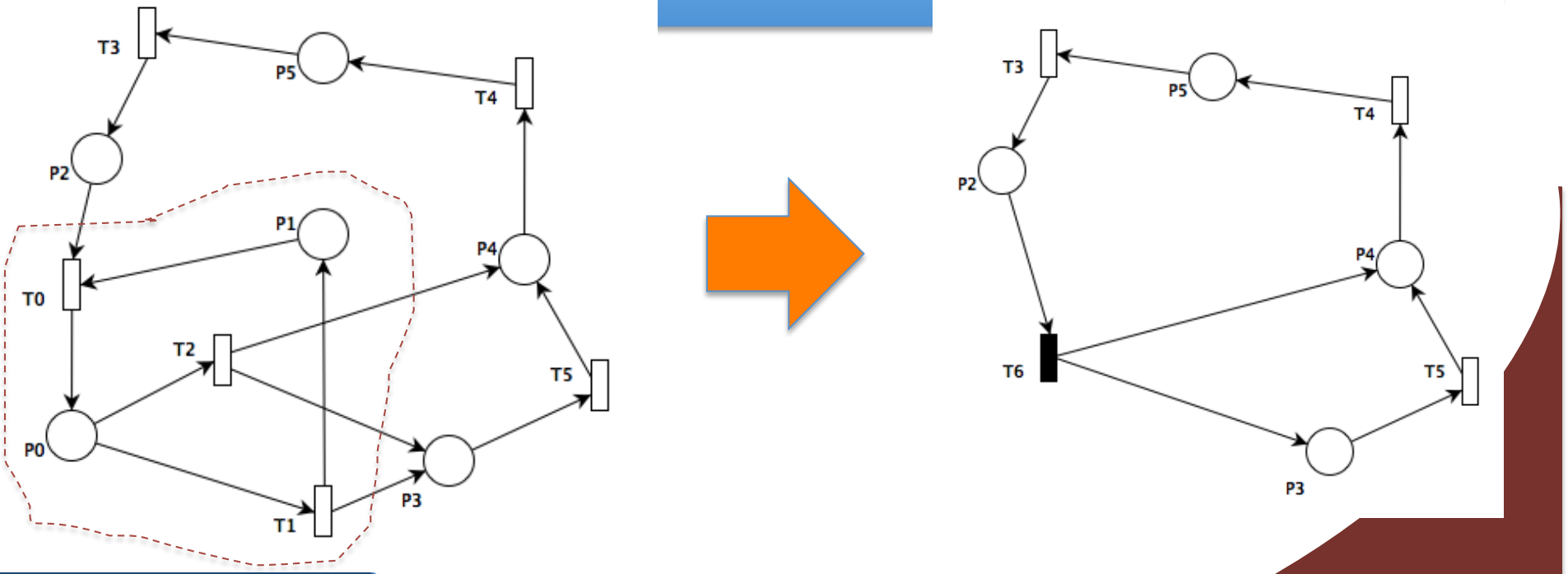
Hierarchy in General

Def. 35] Seja uma estrutura de rede $N = (P,T;F)$. Seja $X = P \cup T$, e um sub - conjunto $Y \subseteq X$. Definimos uma *borda* de N, ao conjunto $\partial(Y) = \{y \in Y \mid \exists x \notin Y \,.\, x \in \mathrm{loc}(y)\}$.

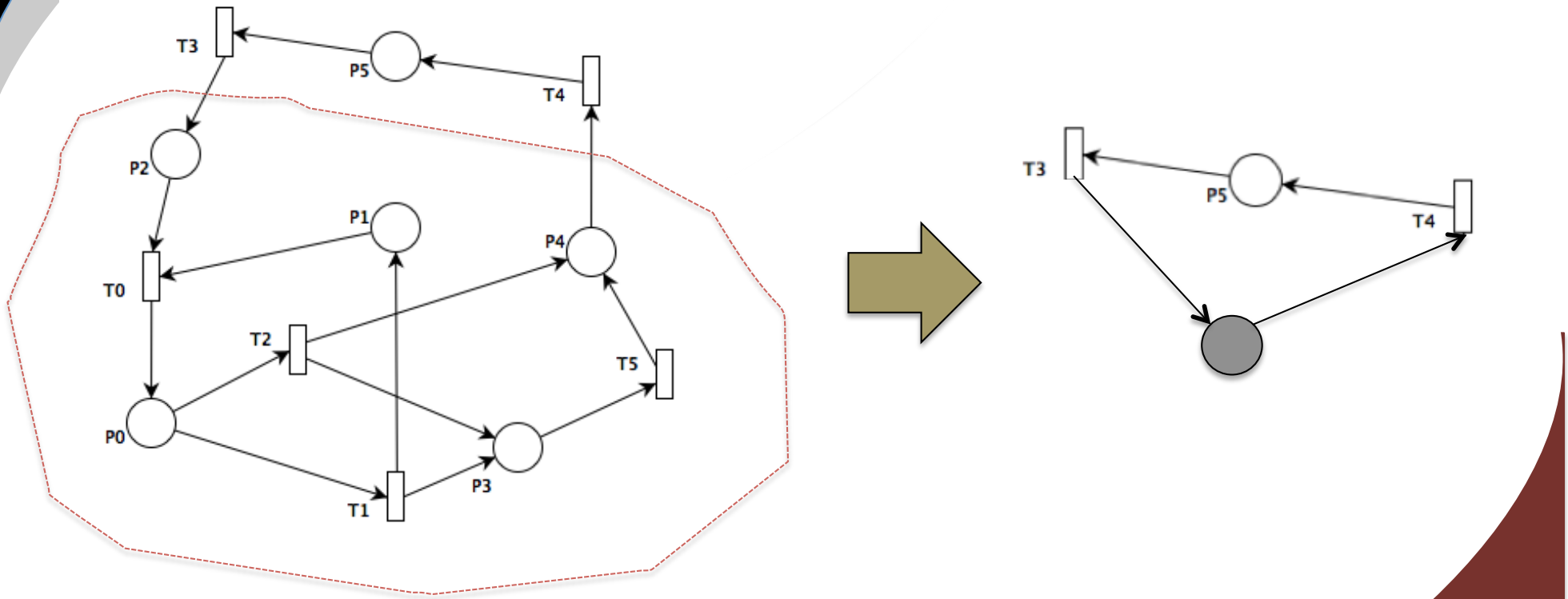


Subnet substitution

Def. 36] Um sub - conjunto de elementos da estrutura de rede $N = (P,T;F)$ é dito *limitado por lugar* (place - bounded) ou aberto se $\partial(Y) \subseteq P$. Este sub - conjunto é dito limitado por transição (transition - bounded) se $\partial(Y) \subseteq T$.



Subnet substitution



The New PN after subs.

Uma nova rede é formada, $N' = (P', T'; F')$ onde (se Y é limitada por transição) :

$P' = P \setminus Y$

$T' = (T \setminus Y) \cup t_y$, *onde* t_y é um novo elemento que substitui a sub-rede Y;

$F' = F \setminus Int(Y)$, onde $Int(Y)$ é o conjunto dos arcos internos de Y.

Similarmente, se Y é limitada por lugar, a nova rede terá,

$P' = (P \setminus Y) \cup p_y$, onde p_y é um novo lugar que substitui a sub-rede Y;

$T' = T \setminus Y$;

$F' = F \setminus Int(Y)$, onde $Int(Y)$ é o conjunto dos arcos internos de Y.

Proper Elements

Seja x_y um elemento genérico (instanciável por t_y ou por p_y). Este elemento é dito *próprio* se e somente se é limitado por transição (lugar), tem somente dois elementos de borda, com pelo menos um processo vivo entre eles.

Se os elementos abstratos são próprios as propriedades da rede subjacente se conservam a menos de um termo aditivo. (J. R. Silva, On The Property Analysis of Abstract and Hierarchical Nets, to appear).

The challenge

Hierarchy is a good abstraction feature. However, the real challenge is to associate that with the property analysis, so that the abstract net preserve the same properties than the expanded one.

The proper requirement is a key issue for that.

Design and abstraction

Substitution transitions work in a similar way as the refinement primitives found in many system description languages – e.g., SADT diagrams.

Fim

Escola Politécnica da USP

MECATRÔNICA