PMR 5237
Modelagem e Design de Sistemas Discretos em Redes de Petri

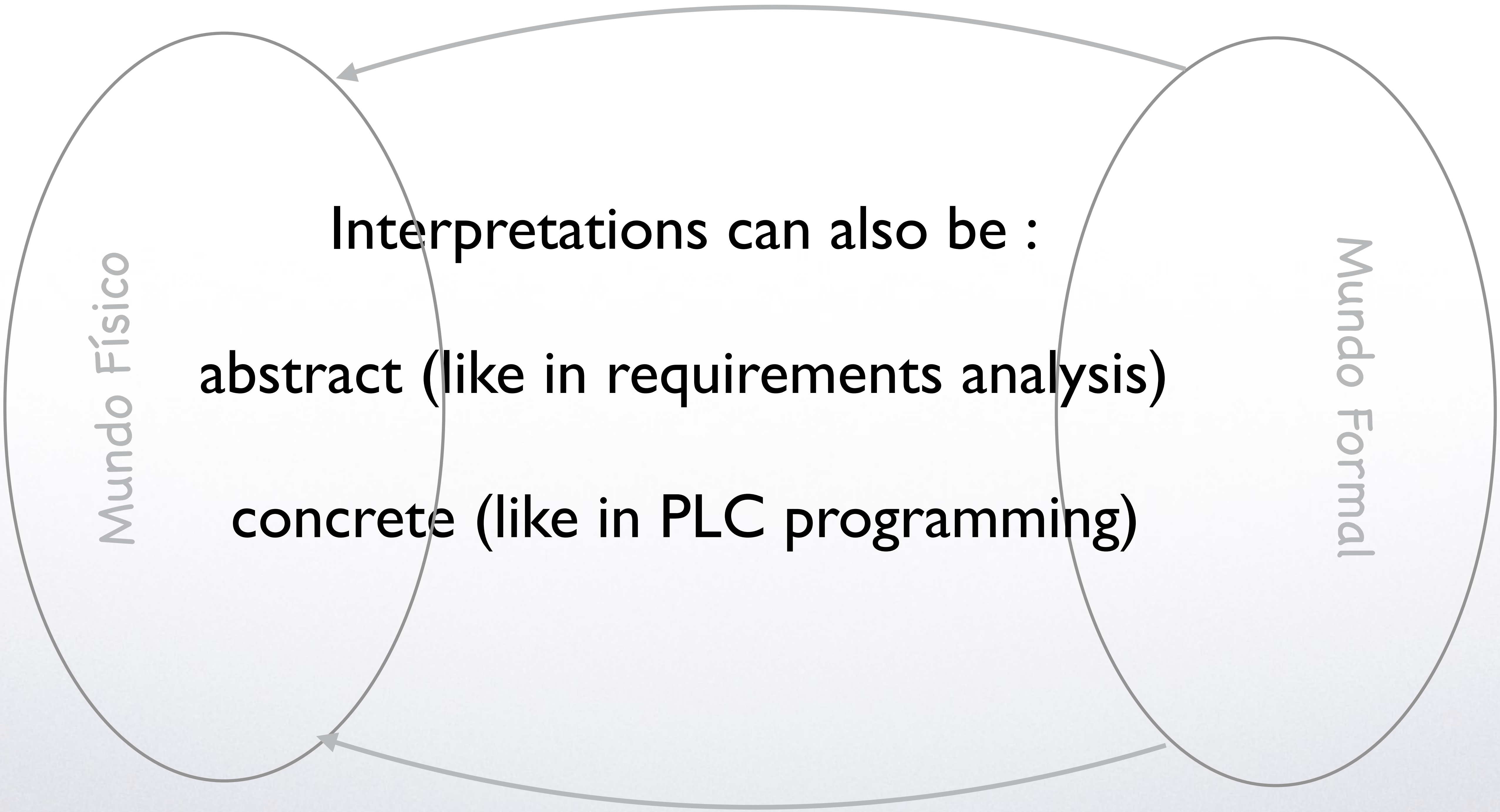
Aula 9 : Análise de Propriedades e Técnicas de Modelagem

Prof. José Reinaldo Silva
reinaldo@usp.br

Modelar com redes clássica

X

Modelar com redes de alto nível

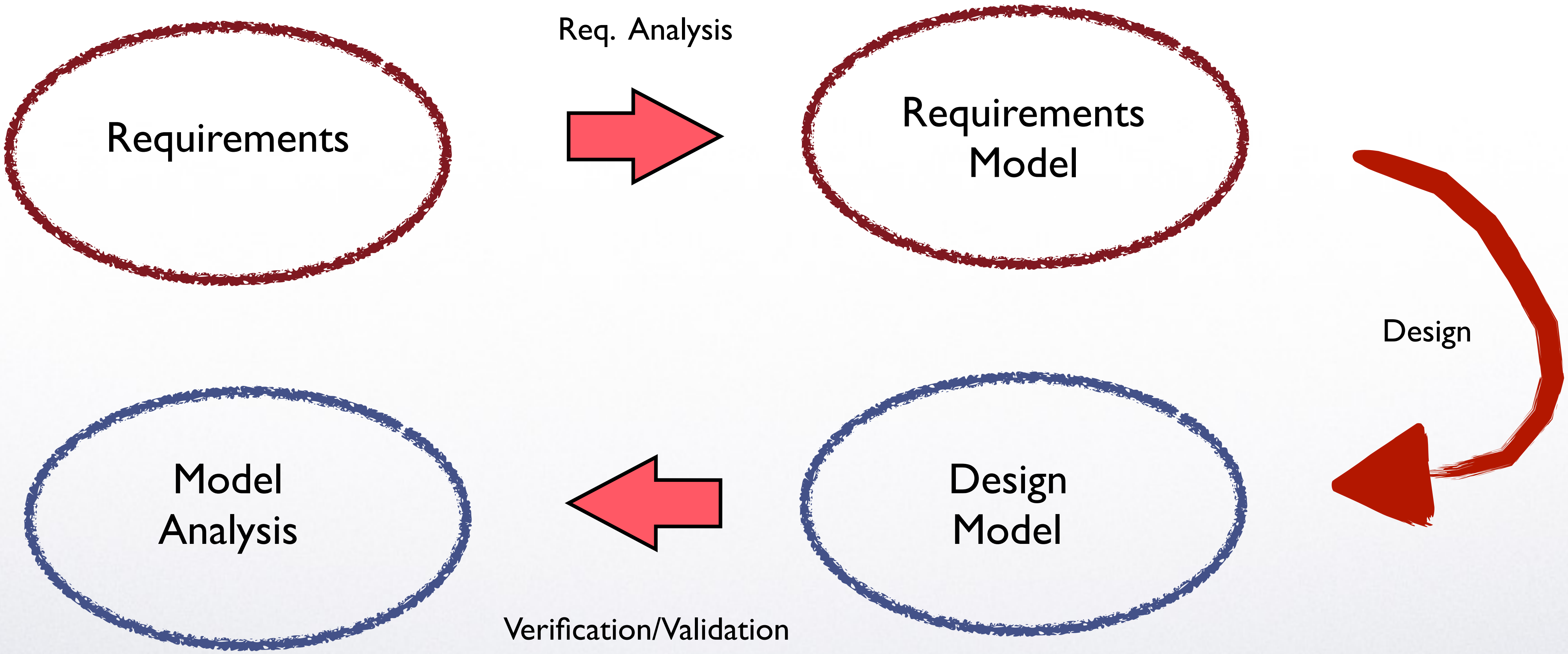


Interpretations can also be :

abstract (like in requirements analysis)

concrete (like in PLC programming)

Use of Petri Nets in Design



d-l@b

MECATRÔNICA

CPN : definição formal

Definition: A Coloured Petri Net is a tuple CPN = (Σ, P, T, A, N, C, G, E, I) satisfying the following requirements:

(i) Σ is a finite set of non-empty types, called **colour sets**.

(ii) P is a finite set of **places**.

(iii) T is a finite set of **transitions**.

(iv) A is a finite set of **arcs** such that:

- $P \cap T = P \cap A = T \cap A = \emptyset$.

(v) N is a **node** function. It is defined from A into $P \times T \cup T \times P$.

(vi) C is a **colour** function. It is defined from P into Σ.

(vii) G is a **guard** function. It is defined from T into expressions such that:

- $\forall t \in T$: $[Type(G(t)) = Bool \wedge Type(Var(G(t))) \subseteq \Sigma]$.

(viii) E is an **arc expression** function. It is defined from A into expressions such that:

- $\forall a \in A$: $[Type(E(a)) = C(p(a))_{MS} \wedge Type(Var(E(a))) \subseteq \Sigma]$

where p(a) is the place of N(a).

(ix) I is an **initialization** function. It is defined from P into closed expressions such that:

- $\forall p \in P$: $[Type(I(p)) = C(p)_{MS}]$.

Modelagem e análise em Redes Colorida/Alto Nível

I'

modelagem em CPN/HLPN → **modelagem em rede P/T**

análise de atingibilidade (simulação)

modelagem em rede P/T:
análise de atingibilidade (simulação)
análise de invariantes
análise de performance

I

modelagem em rede P/T → **modelagem em CPN/HLPN**

análise de atingibilidade (simulação)
análise de invariantes
análise de performance

modelagem em CPN/HLPN:
análise de atingibilidade (simulação)

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.

Correct interpretation means that the "behavior" of practical system (described in some language) is a metaphor for the high level net (syntactically and semantically).

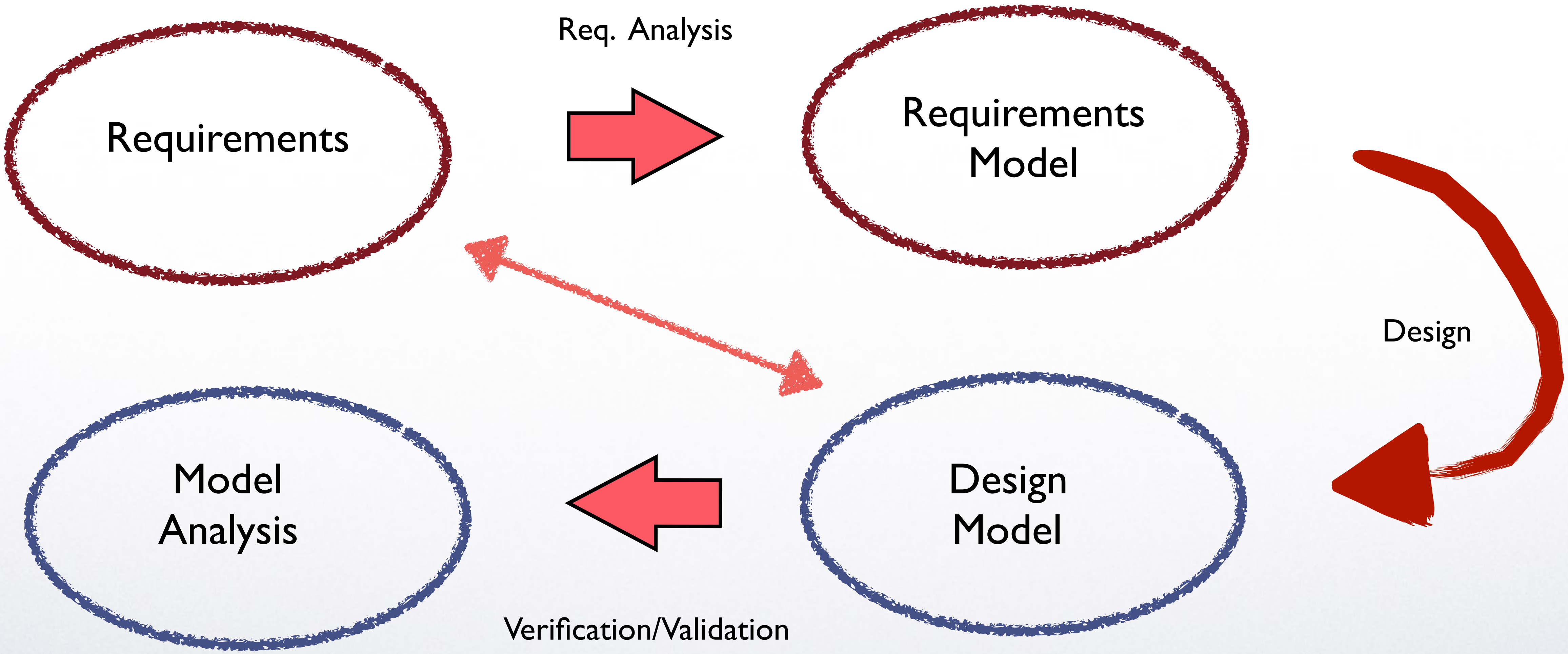
Even if environments can be an apprentice for that there is no guarantee to get a good metaphor. Only going through the modelling, analysis and verification we can be certain about "correctness".

Interpretations can also be :

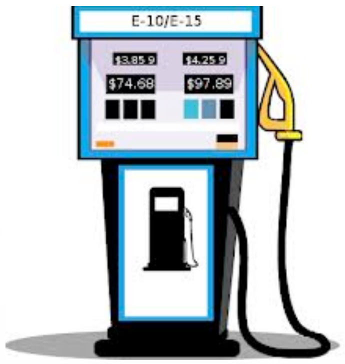
abstract (like in requirements analysis)

concrete (like in PLC programming)

Use of Petri Nets in Design



A concrete example

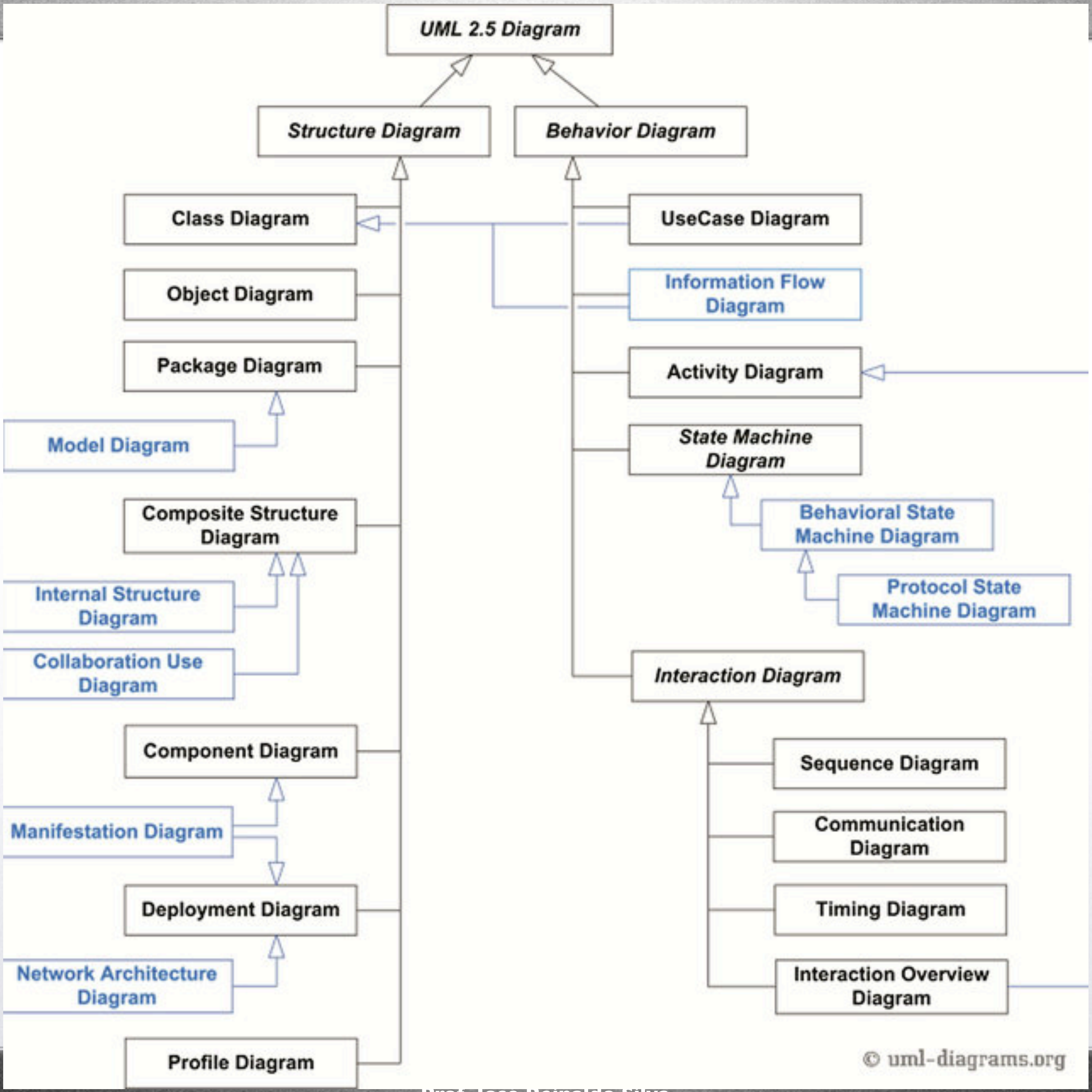


Exit

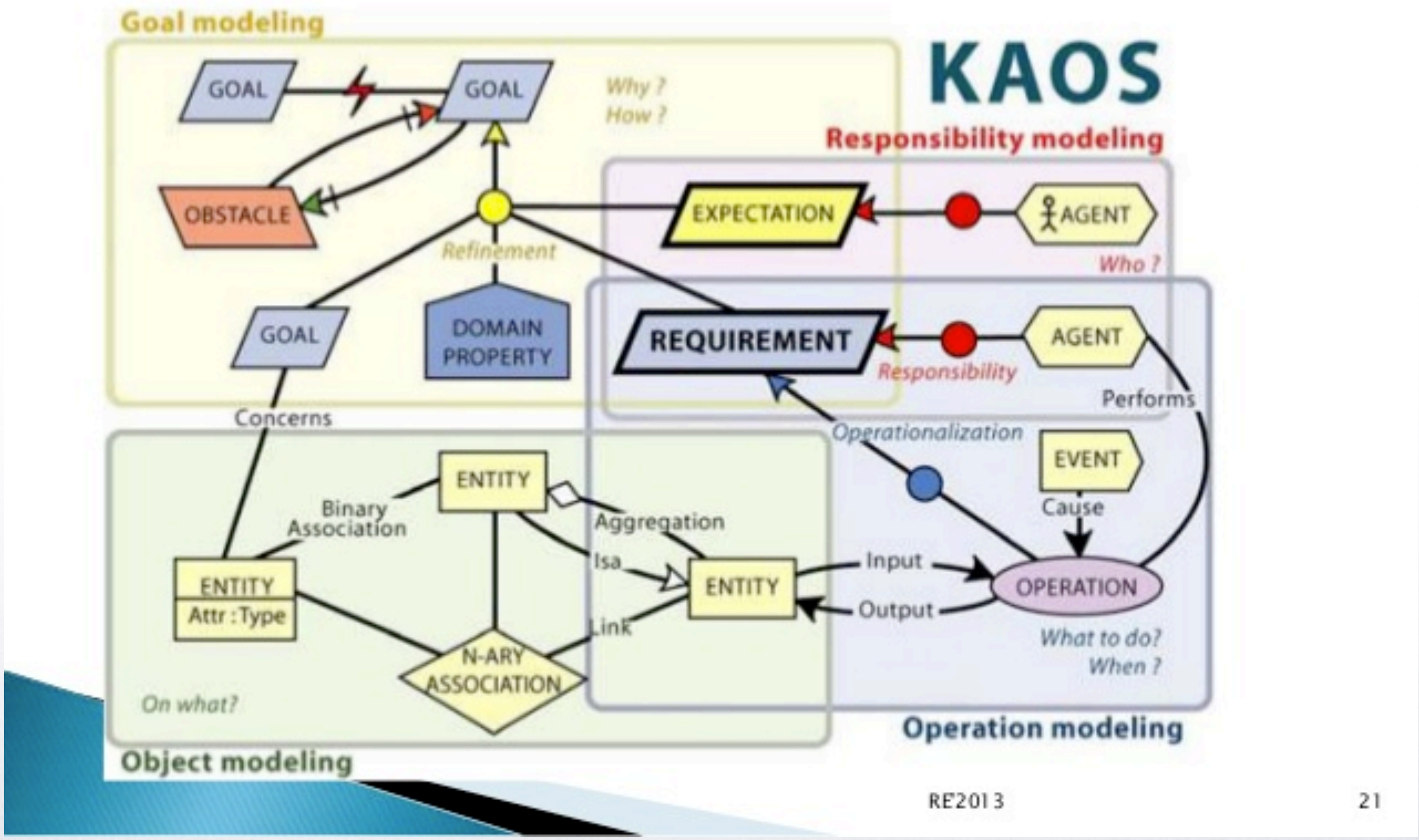
MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.


UML 2.5 Diagram
Structure Diagram
Behavior Diagram
Class Diagram
Object Diagram
Package Diagram
Model Diagram
Composite Structure Diagram
Internal Structure Diagram
Collaboration Use Diagram
Component Diagram
Manifestation Diagram
Deployment Diagram
Network Architecture Diagram
Profile Diagram
UseCase Diagram
Information Flow Diagram
Activity Diagram
State Machine Diagram
Behavioral State Machine Diagram
Protocol State Machine Diagram
Interaction Diagram
Sequence Diagram
Communication Diagram
Timing Diagram
Interaction Overview Diagram
© uml-diagrams.org

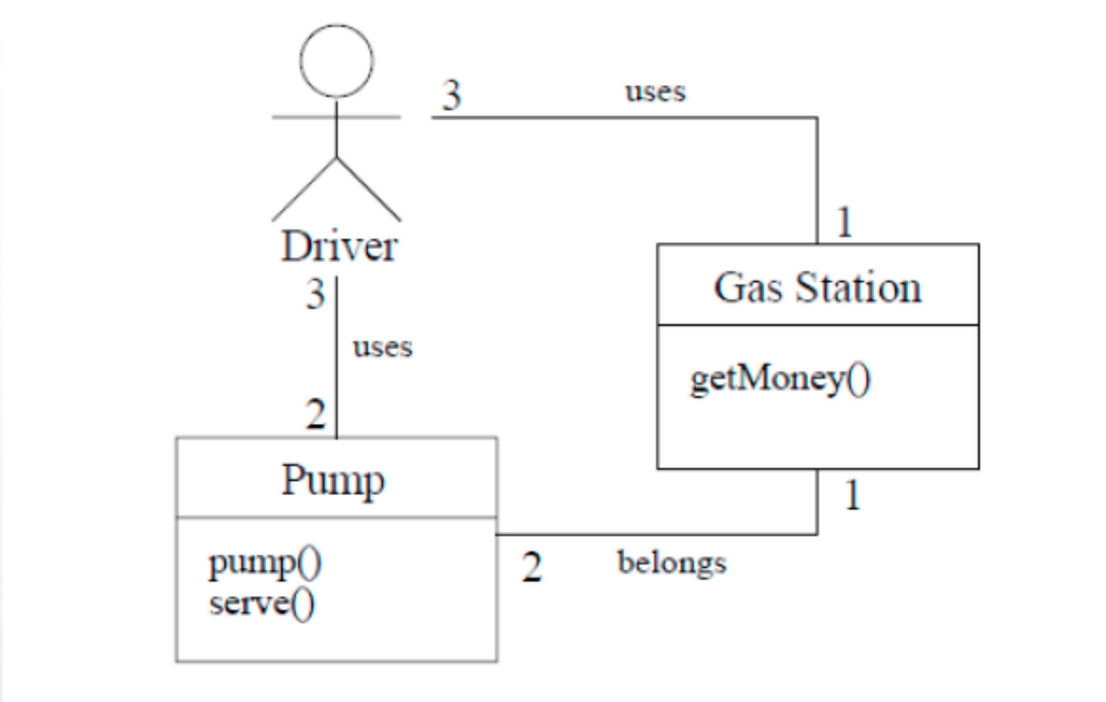
KAOS main model elements

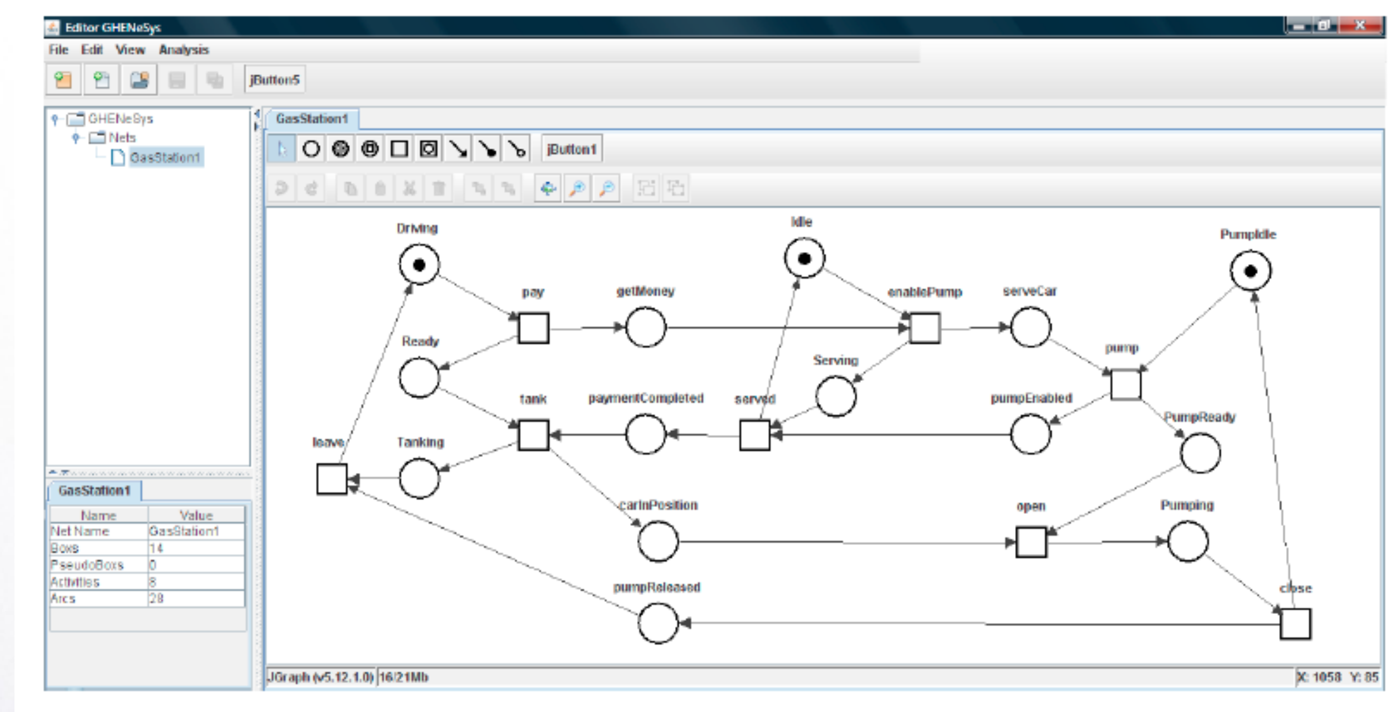


Queremos automatizar quase completamente (esta é uma abertura do projeto, ter alguém para receber o pagamento ou usar cartão somente) o serviço de atendimento em um posto de gasolina. Os sub-serviços seriam o abastecimento de combustível, gás, óleo (calibração seria gratuita). Precisamos portanto fazer o design do "work flow" para que os clientes não tenha que esperar em filas, e para que o posto não tenha prejuízo financeiro. Trata-se portanto de distribuir os clientes nos serviços em ordem diferente (como no exemplo da fábrica flexível) para obter estes objetivos.

The Gas Station Problem

del Foyo, P.M.G., Salmon, A.O., Silva, J.R.; Requirements Analysis of Automated Problems Using UML/Petri Nets, Proc. of the 21st. Congress of Mech. Eng., Natal, 2011.





Baresi, L. and Pezze, M., 2001. "Improving UML with petri nets". In *UNIGRA 2001, Uniform Approaches to Graphical Process Specification Techniques (a Satellite Event of ETAPS 2001)*. Elsevier, Vol. 44 of *Electronic Notes in Theoretical Computer Science*, pp. 107–119.

Available online at www.sciencedirect.com

ScienceDirect

ELSEVIER

IFAC Papers Online
CONFERENCE PAPER ARCHIVE

IFAC-PapersOnLine 48-3 (2015) 1634–1639

Combining KAOS and GHENeSys in the requirement and analysis of service manufacturing

Javier Martinez Silva * José Reinaldo Silva **

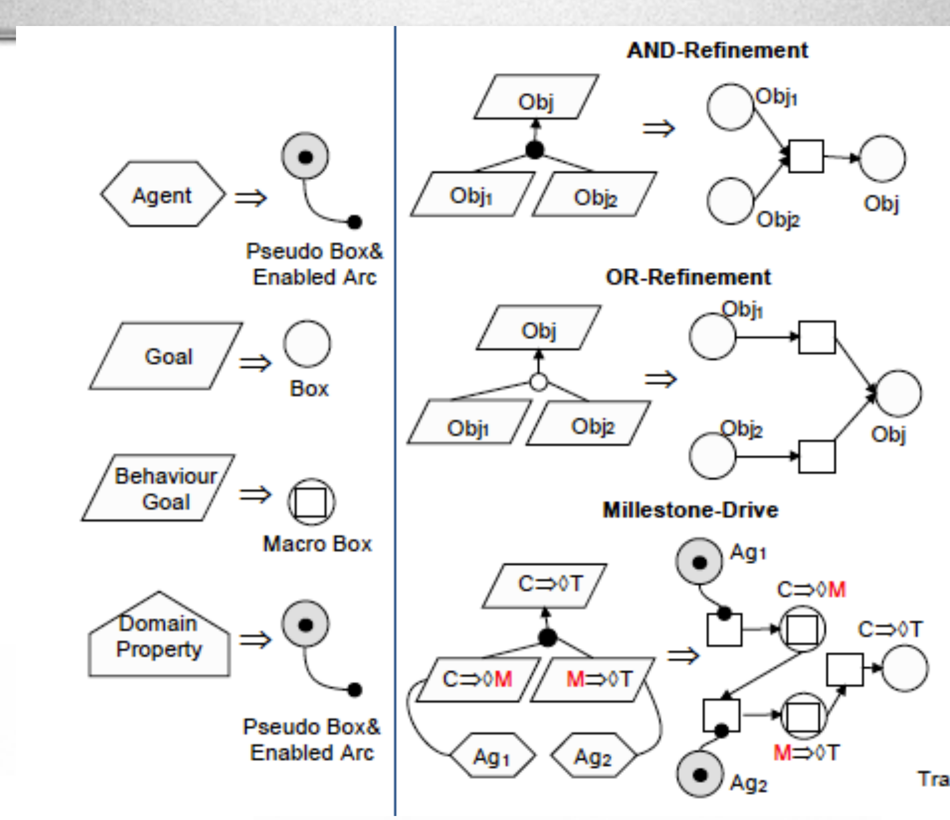
* *Department of Mechatronics Engineering, University of São Paulo, São Paulo, Brazil, (e-mail: javsilva@usp.br).*
** *Department of Mechatronics Engineering, University of São Paulo, São Paulo, Brazil, (e-mail: reinaldo@usp.br).*

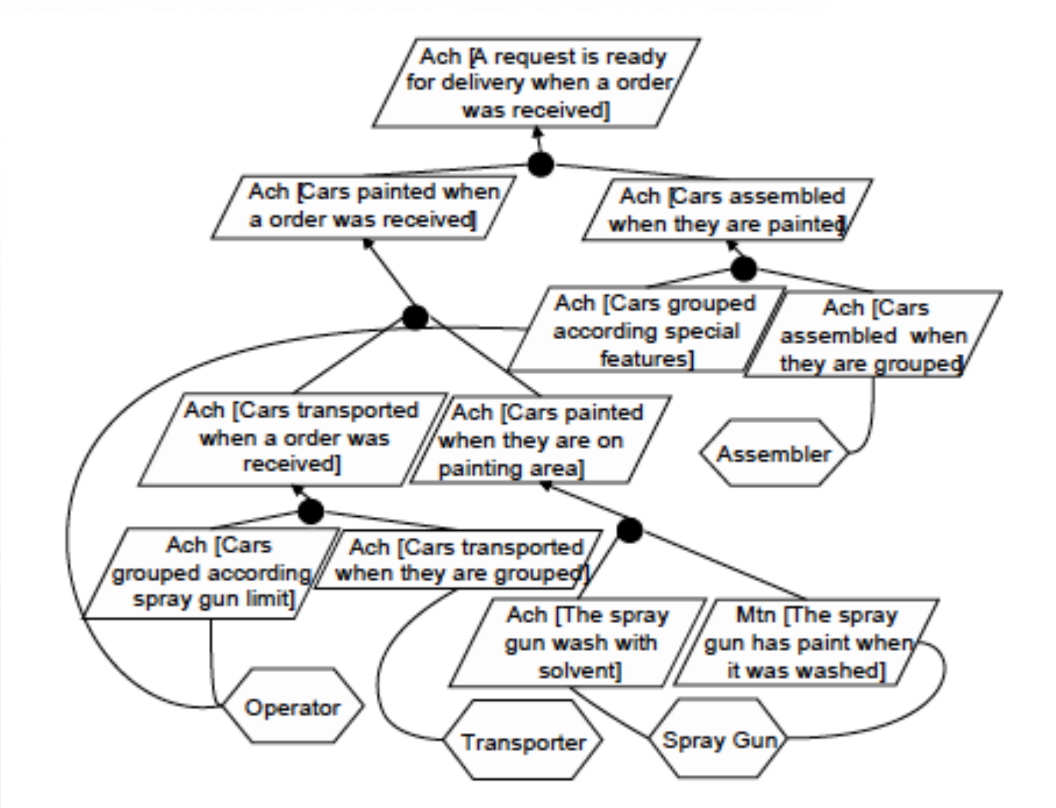
Abstract: The design of planning problems has demanded more attention from the academy - which looks for new approaches - and from practitioners - which seeks for new and better results for real applications. AI approaches have been used for different applications in many domains, including manufacturing, and specially manufacturing services. However, as in all domains, a design process has to be provided to the planning (and scheduling) design to get all advantages in the use of heuristic or analytic methods. Requirements analysis is a key issue for that design process and needs to be enhanced to fit users and stakeholder expectations. In this work we focus on requirements analysis for planning problems using Petri Nets derived from a UML representation of requirements. A transference algorithm is proposed to synthesize a Petri Net from KAOS diagram and allow analysis using Petri Net property analysis suitable to be applied to large projects.

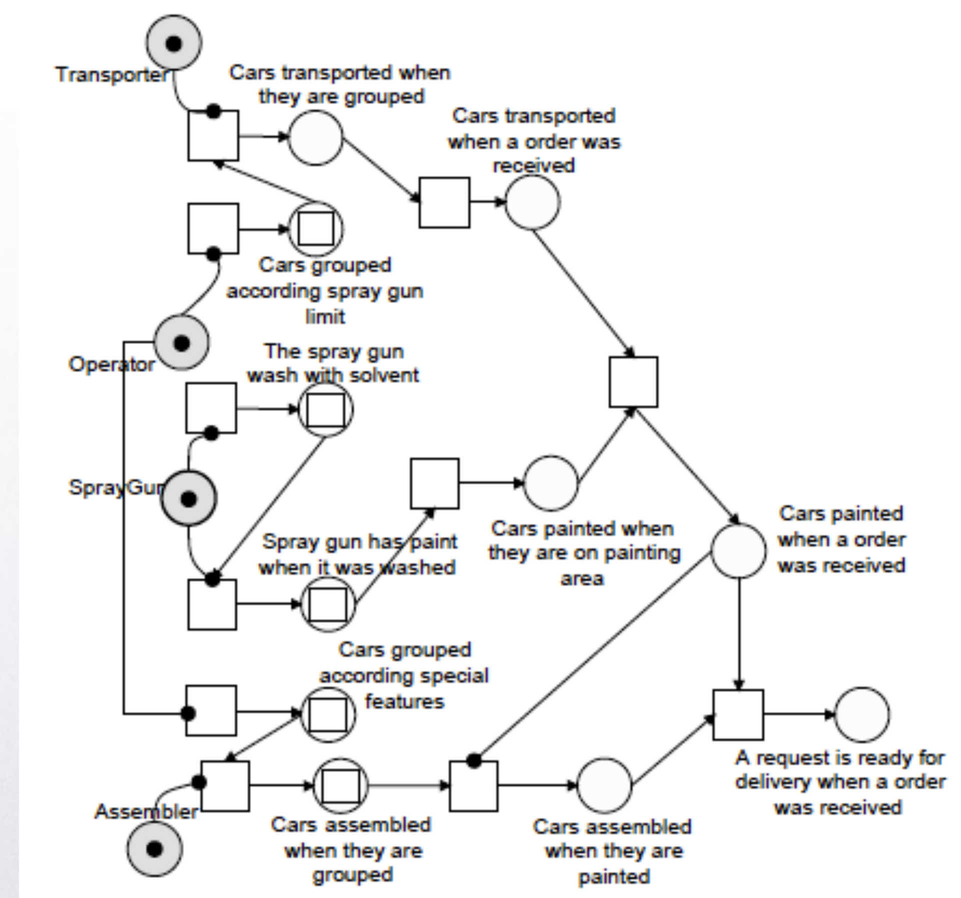
Keywords: manufacturing services, requirements analysis, KAOS, Petri Nets.

1. INTRODUCTION

depend on the requirement specifications and divided

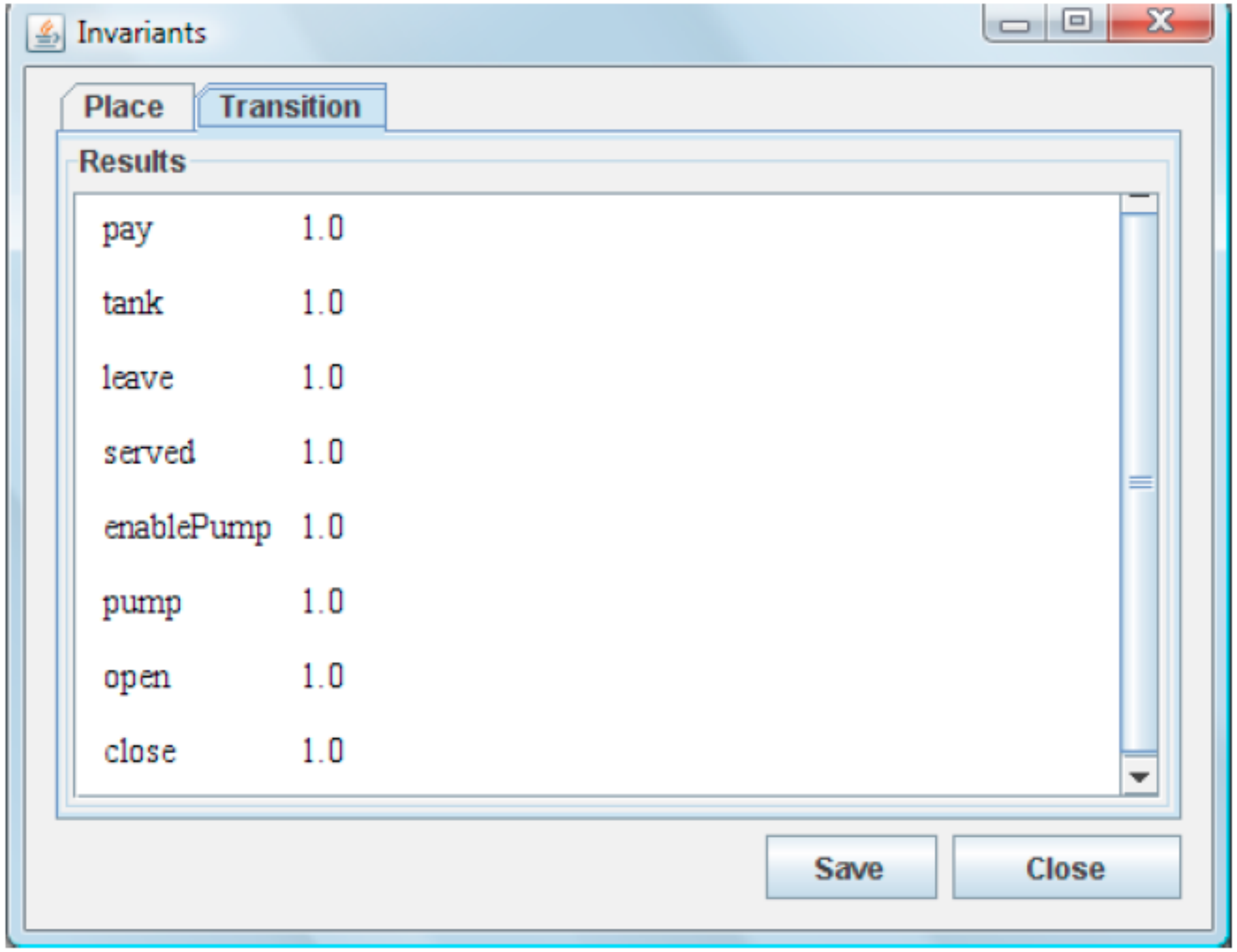

Agent
Pseudo Box& Enabled Arc
Goal
Box
Behaviour Goal
Macro Box
Domain Property
Pseudo Box& Enabled Arc
AND-Refinement
Obj
Obj1
Obj2
OR-Refinement
Millestone-Drive
C⇒◊T
C⇒◊M
M⇒◊T
Ag1
Ag2

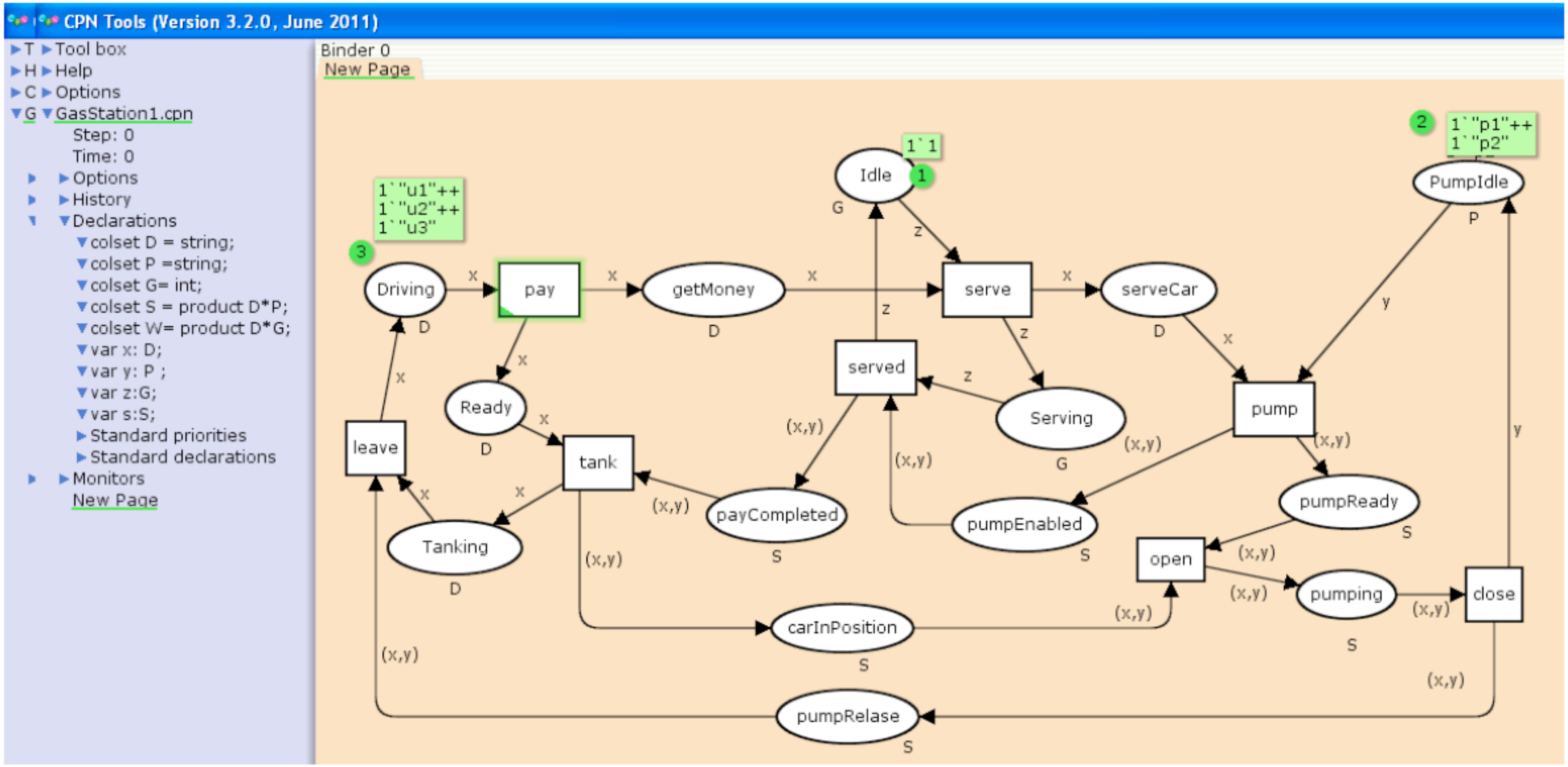

Ach [A request is ready for delivery when a order was received]
Ach [Cars painted when a order was received]
Ach [Cars assembled when they are painted]
Ach [Cars grouped according special features]
Ach [Cars assembled when they are grouped]
Ach [Cars transported when a order was received]
Ach [Cars painted when they are on painting area]
Assembler
Ach [Cars grouped according spray gun limit]
Ach [Cars transported when they are grouped]
Ach [The spray gun wash with solvent]
Mtn [The spray gun has paint when it was washed]
Operator
Transporter
Spray Gun

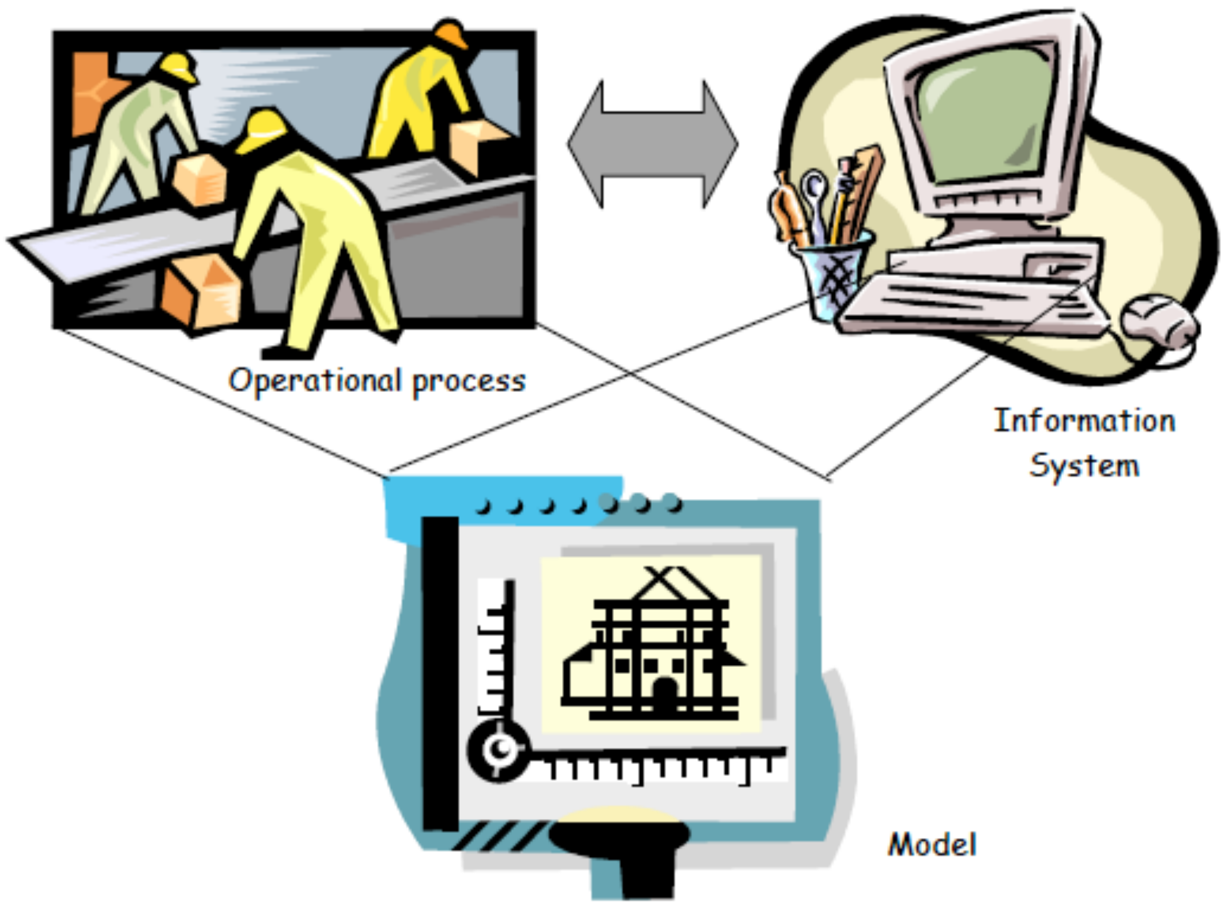

Transporter
Cars transported when they are grouped
Cars transported when a order was received
Cars grouped according spray gun limit
Operator
The spray gun wash with solvent
SprayGun
Spray gun has paint when it was washed
Cars painted when they are on painting area
Cars painted when a order was received
Cars grouped according special features
Assembler
Cars assembled when they are grouped
Cars assembled when they are painted
A request is ready for delivery when a order was received

Requirements Analysis

All transitions must fire to get a complete cycle for using the gas station (it does not matter how many Pumps it has)




CPN Tools (Version 3.2.0, June 2011)
T ▶ Tool box
H ▶ Help
C ▶ Options
G ▼ GasStation1.cpn
Step: 0
Time: 0
Options
History
Declarations
colset D = string;
colset P =string;
colset G= int;
colset S = product D*P;
colset W= product D*G;
var x: D;
var y: P ;
var z:G;
var s:S;
Standard priorities
Standard declarations
Monitors
New Page
Binder 0
New Page
1`"u1"++
1`"u2"++
1`"u3"
1`1
1`"p1"++
1`"p2"
Driving
pay
getMoney
Idle
serve
serveCar
PumpIdle
Ready
served
Serving
pump
leave
tank
payCompleted
pumpEnabled
pumpReady
Tanking
open
pumping
close
carInPosition
pumpRelase



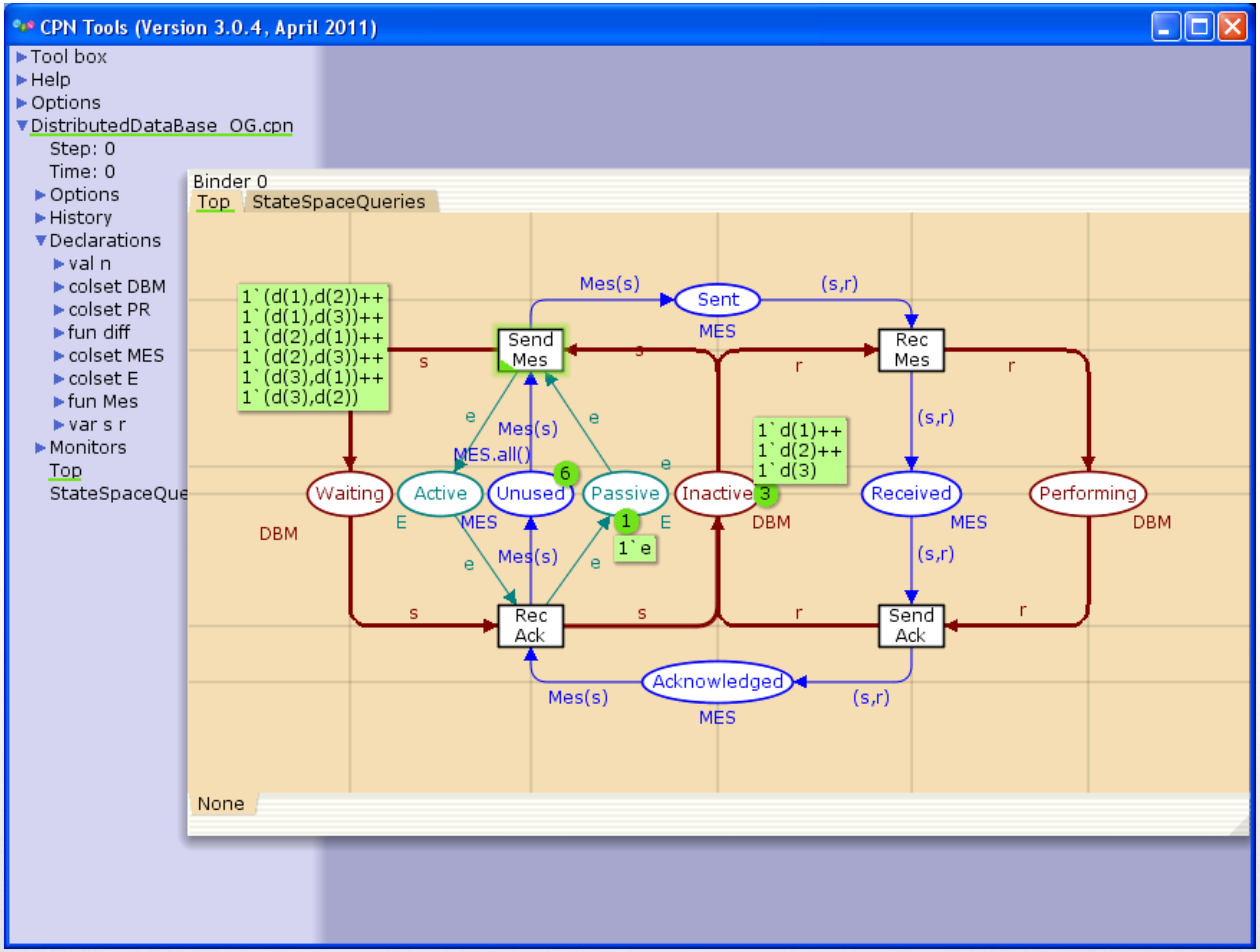
- Analysis is typically model-driven to allow e.g. what-if questions.
- Models of both operational processes and/or the information systems can be analyzed.
- Types of analysis:
  - ***validation***
  - ***verification***
  - ***performance analysis***

1. Reachability/coverability graph
2. Structural techniques
   - Place and transition invariants
   - Marking equation
   - Traps, siphons, etc.
3. Simulation

- Each can be applied to both classical and high-level Petri nets.
- Nevertheless, for the second we restrict ourselves to classical Petri nets.

Mapping technique/use:

- reachability graph (validation, verification)
- invariants (validation, verification)
- simulation (validation, performance analysis)


CPN Tools (Version 3.0.4, April 2011)
Tool box
Help
Options
DistributedDataBase_OG.cpn
Step: 0
Time: 0
Options
History
Declarations
val n
colset DBM
colset PR
fun diff
colset MES
colset E
fun Mes
var s r
Monitors
Top
StateSpaceQue
Binder 0
Top
StateSpaceQueries
1`(d(1),d(2))++
1`(d(1),d(3))++
1`(d(2),d(1))++
1`(d(2),d(3))++
1`(d(3),d(1))++
1`(d(3),d(2))
Mes(s)
Sent
MES
(s,r)
Send Mes
Rec Mes
s
r
e
Mes(s)
MES.all()
(s,r)
1`d(1)++
1`d(2)++
1`d(3)
Waiting
Active
Unused
Passive
Inactive
Received
Performing
6
3
1
DBM
E
MES
E
DBM
MES
DBM
1`e
Mes(s)
Rec Ack
Send Ack
Acknowledged
Mes(s)
(s,r)
MES
None

CPN Analysis

CPN Analysis follow the same procedures than the classical analysis and face the same problems, even if has new formal resources to include.

Again, we have a state/transition approach, in a discrete flavor, with the possibility of explosion of the number of combinations of states, that is, in the composition of processes.

Modelagem clássica

As redes possuem propriedades típicas dos esquemas que as tornam
Uma excelente representação formal para sistemas (dinâmicos) discretos,
Entre os quais figuram :

o princípio da dualidade
o princípio da localidade
o princípio da concorrência
o principio da representação gráfica
o princípio da representação algébrica

Special configurations

As in the classic net systems, we detached specific net configurations that constitute a challenge in the analysis process (or situations to be avoided, in order to have the desirable system). Some of this situations are

- Conflict
- Branching and synchronization
- Deadlocks

Distribution and concurrency

As before, everything is based on the concept of locality. According to that, individual states could be classified as independent, and, in such a case, they could be grouped in macro states called cases. Conversely, independent transitions could be also grouped in steps.

Thus, cases and steps could be arranged in a dual way, generating a more abstract net.

Classic and HL modelling

In fact, modelling in Classic P/T nets is very similar to modelling in CPH or other HL net. However, the including of type theory and consequently distinguished marking should not be underestimate, specially to model complex systems. This is the feature that open the possibility to apply Petri nets to more (and different domais.

Extensions

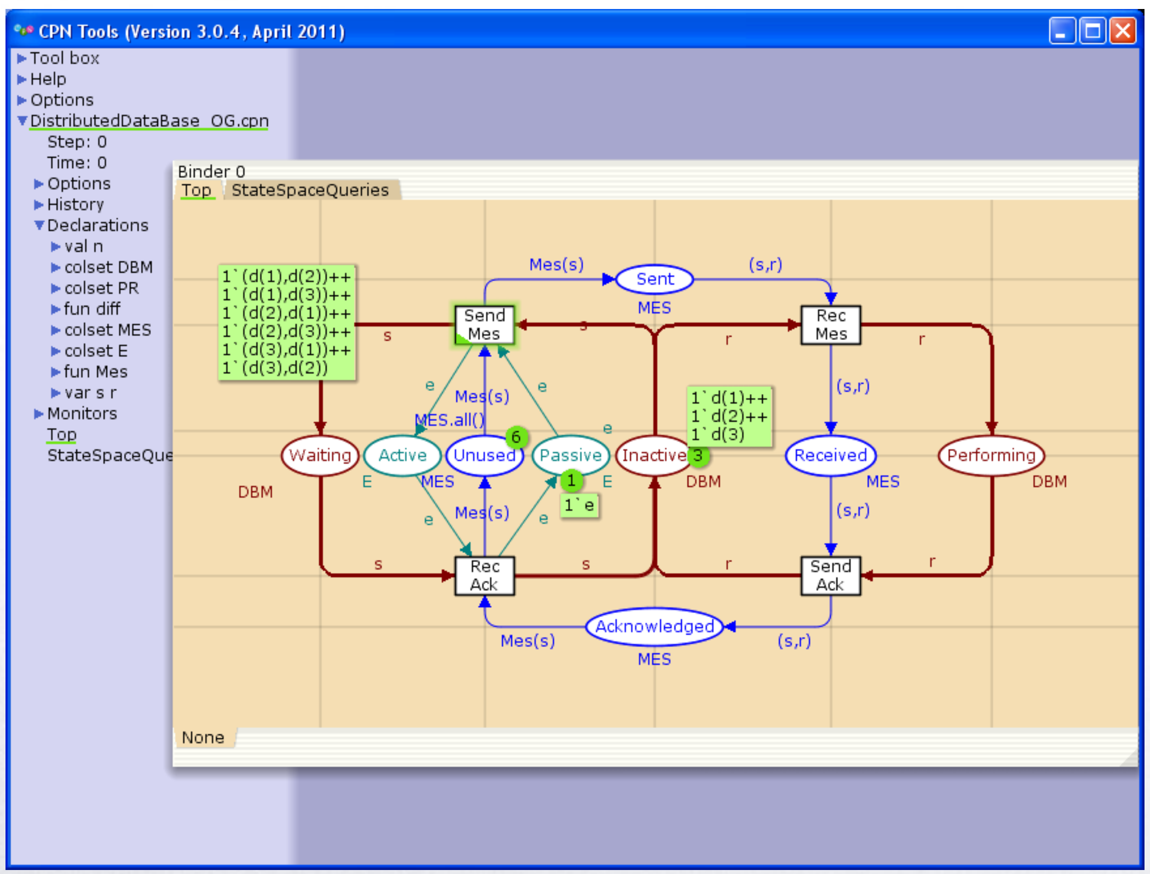
Even extension elements are used in a similar way. So far we have seen two very important extension elements:

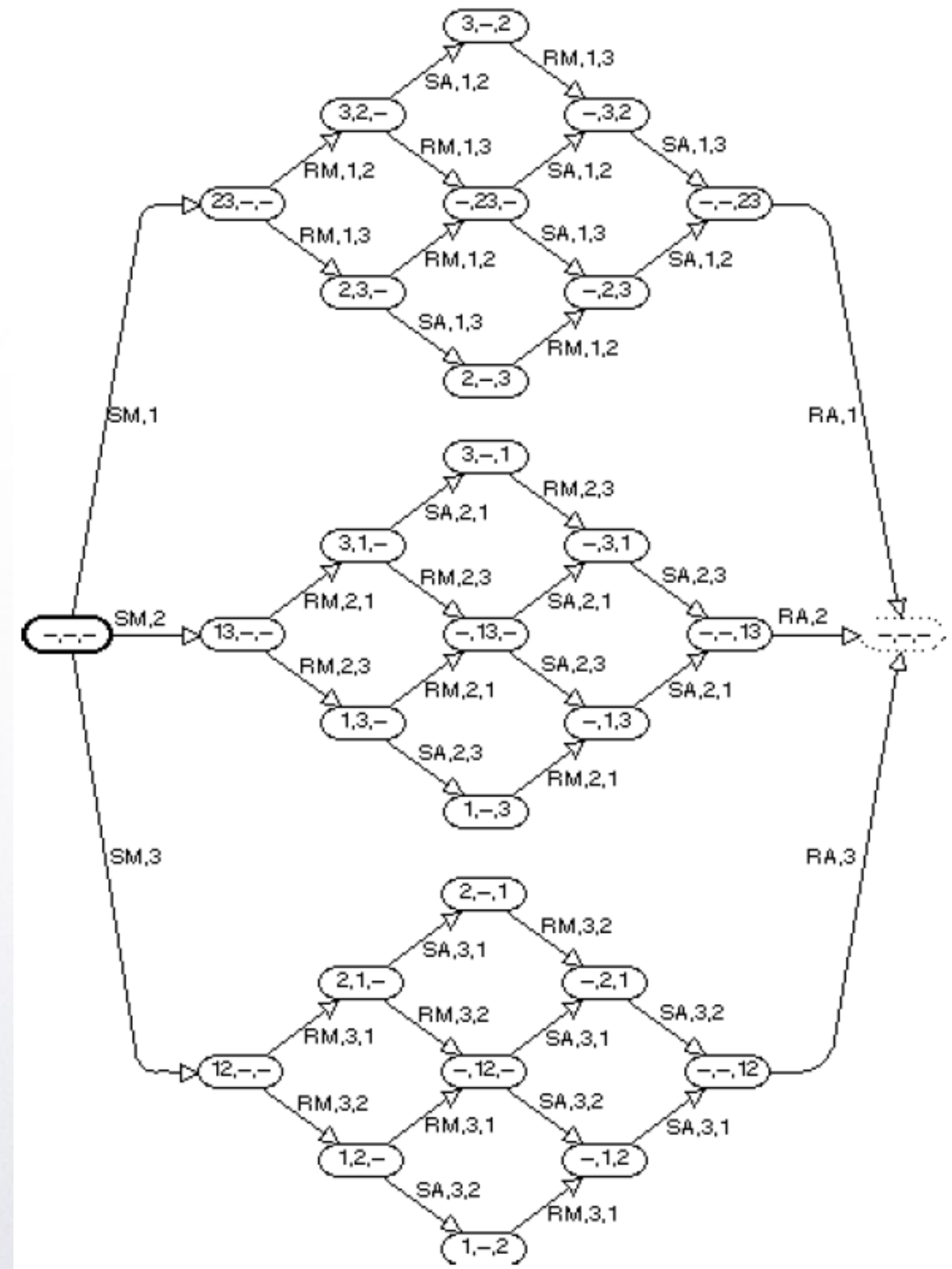
- Gates

- Hierarchy

→ Objects

Analysis in CPN Nets





Directed Graphs

Definition 37

A direct graph is 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 or mapping $A \rightarrow V \times V$.

DG is finite if V and A are finite.

O-Graph

Definition 38

The full ocurrence graph of a CP-net, also called O-graph, is a directed graph $OG = \{V, A, N)$ where:
i) $V = |M_0\rangle$;
ii) $A = \{(M_1, b, M_2) \in (V \times BE \times V) \,|M_1\, |b\rangle\, M_2\}$,
iii) $\forall (M_1, b, M_2) \in A\, N(M_1, b, M_2) = (M_1, M_2)$.

O Graph algorithm

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.
```

The invariant 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 the 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.

The invariant method 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).

Invariants are a very important feature in CPN Design. However, we should not expect to solve the design problem by just inserting invariant analysis.

Besides those inherent problems with invariants, the difficulty to apply this approach to large systems is still present.

CP-nets may be large

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

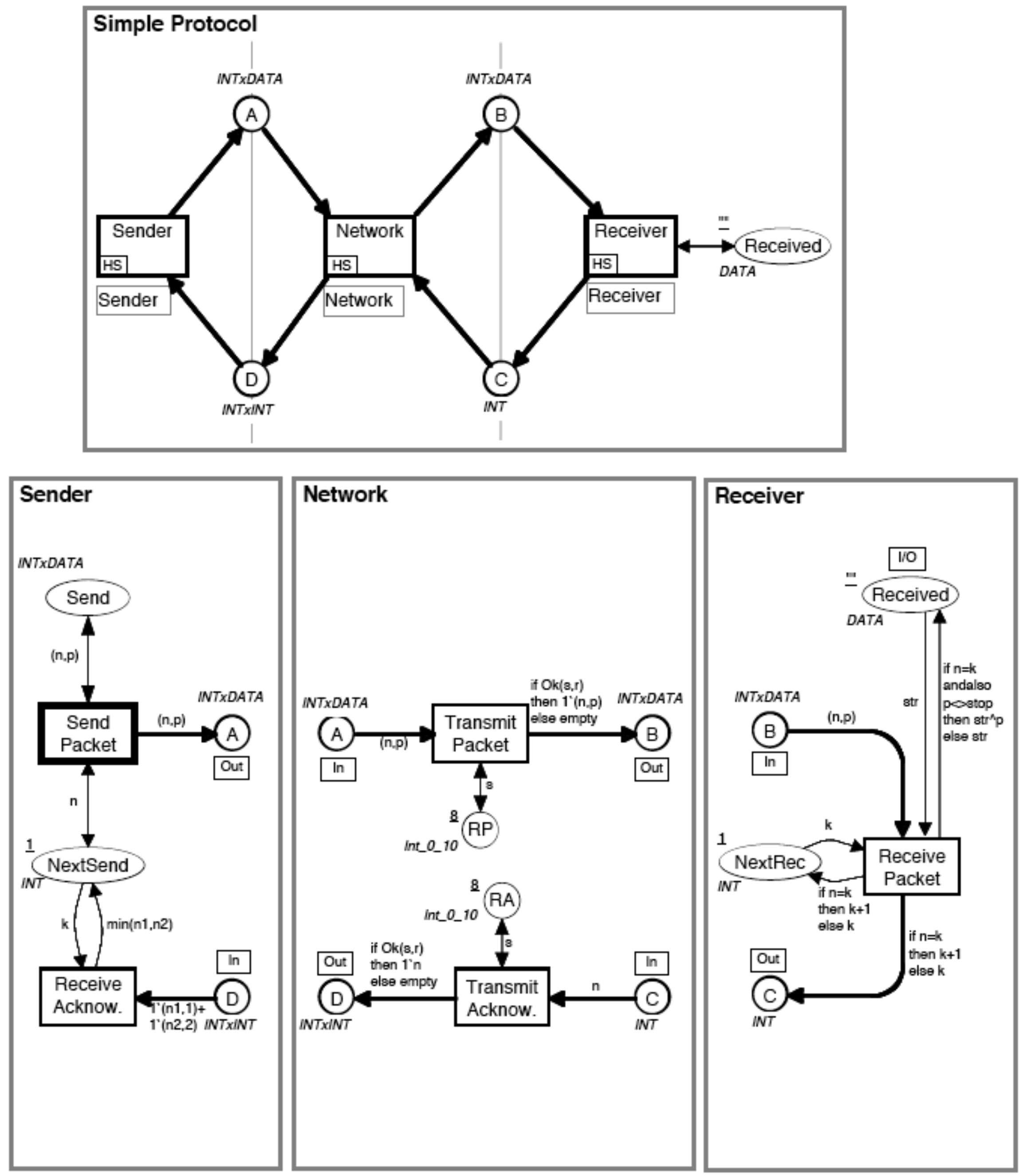
- 10-200 *pages*.
- 50-1000 *places and transitions*.
- 10-200 *colour sets*.

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.

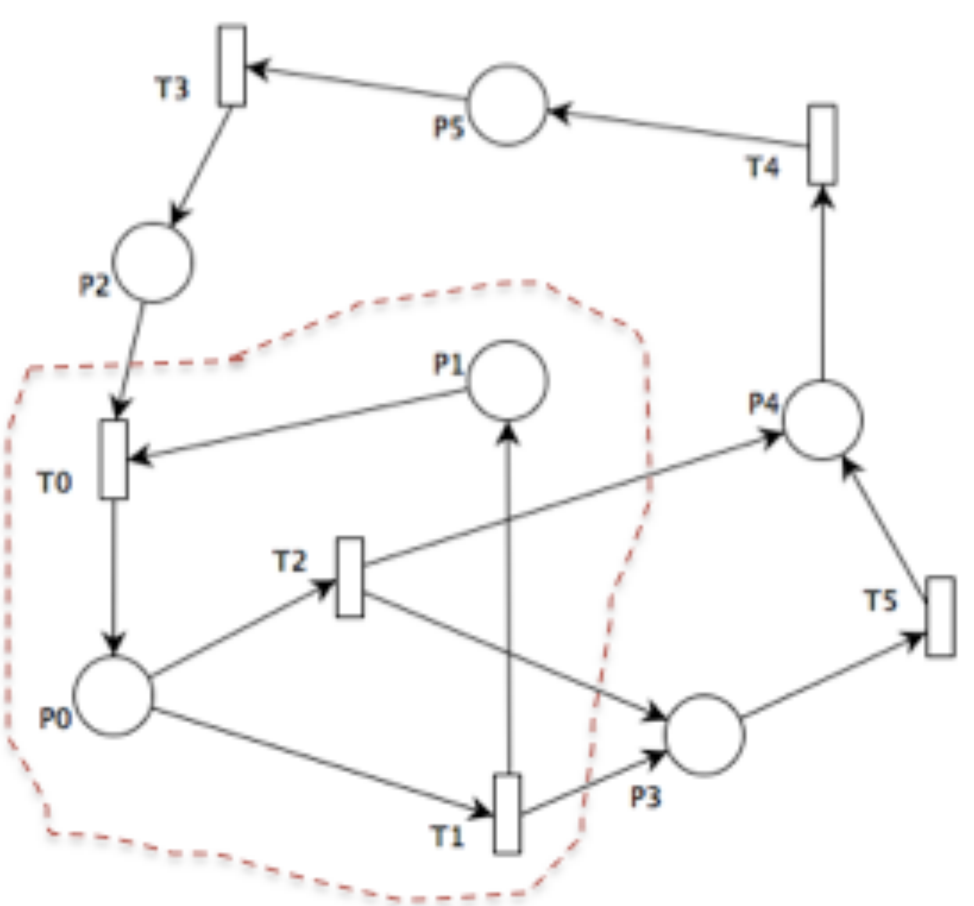
A hierarchical CP-net contains a number of *interrelated subnets*– called *pages*.



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.

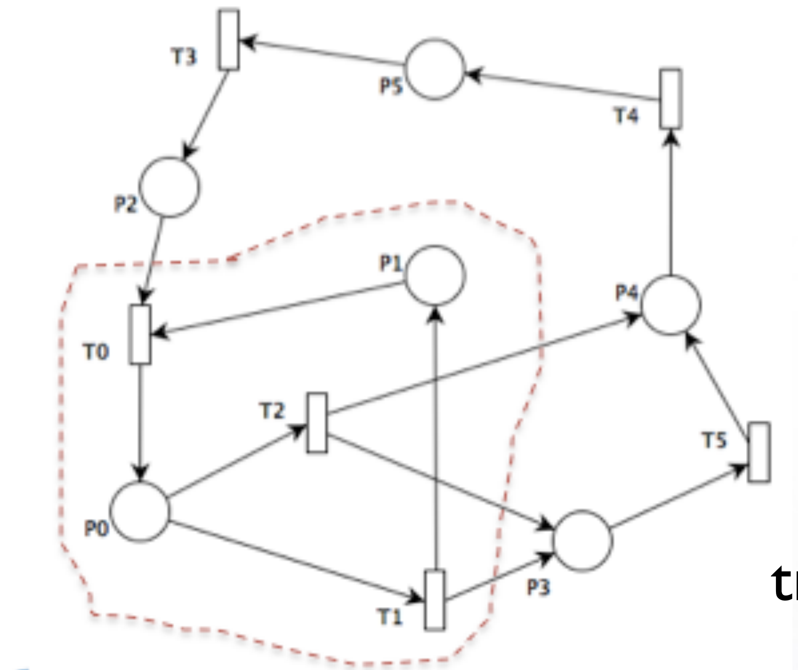
Hierarquia em redes clássicas



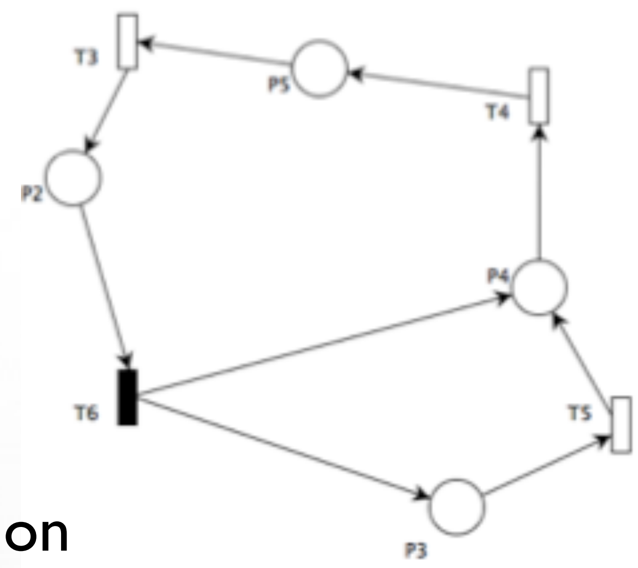
Definition 39

Seja uma estrutura de rede $N = (S, T; F)$. Seja $X = S \cup T$ e um sub-cojunto $Y \subseteq X$. Definimos uma borda de N como o conjunto $\partial(N) = \{y \in Y \mid \exists x \notin Y . x \in loc(y)\}$.

Substituição de uma sub-rede

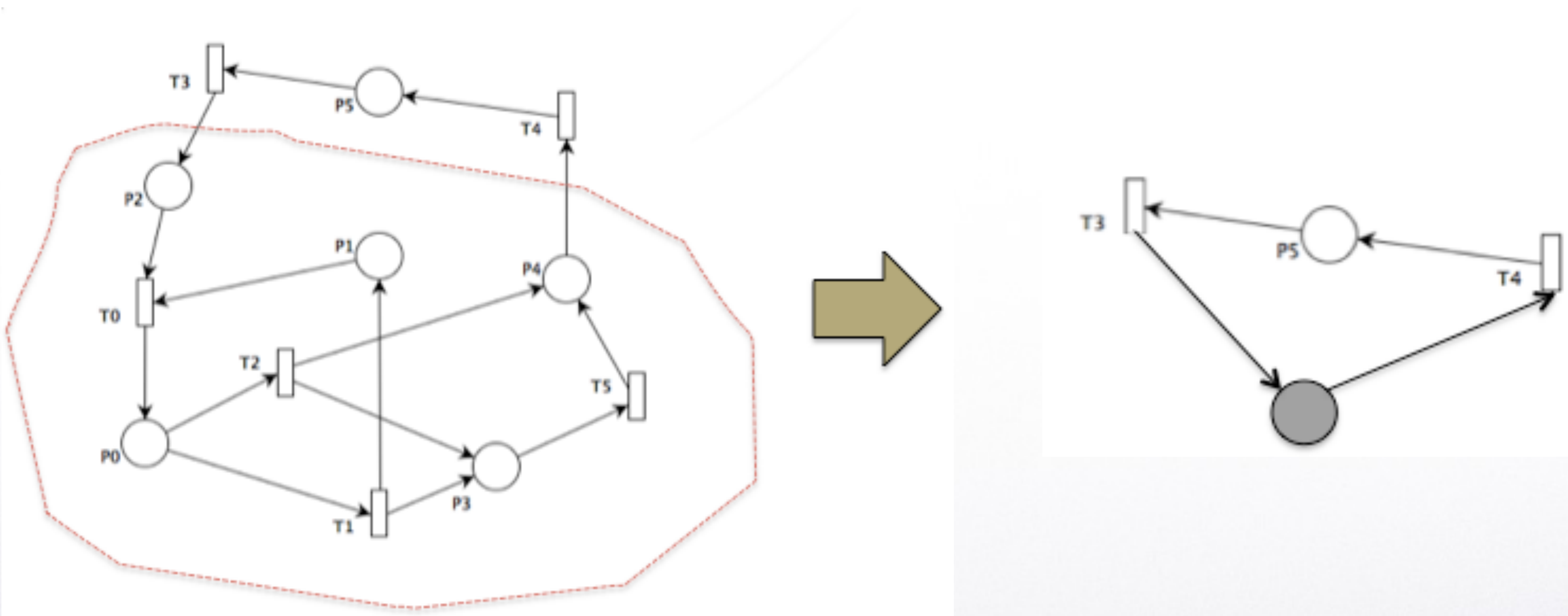


transition bounded substitution



Definition 40

Um sub-conjunto de elementos Y da rede $N = (S, T; F)$ é dito limitado por lugar (place bounded) ou aberto, se e somente se $\partial(Y) \subseteq S$. Similarmente, um sub-conjunto Y desta rede é dito limitado por transição (transition bounded), se e somente se $\partial(Y) \subseteq T$.



place bounded substitution

Se em uma rede com estrutura $N = (S, T; F)$ existe uma sub-rede Y limitada por transição, a substituição desta sub-rede Y gera uma rede $N' = (S', T'; F')$ onde:

(i) $S' = S \setminus Y$;
(ii) $T' = T \setminus Y \cup \{t_y\}$, onde t_y é o novo elemento que substitui Y;
(iii) $F' = F \setminus Int(Y)$ onde $Int(Y)$ é o conjunto dos arcos internos de Y.

.

Similarmente, se a sub-rede Y é limitada por lugar,
(i) $S' = S \setminus Y \cup \{s_y\}$, onde s_y é o novo elemento que substitui Y;
(ii) $T' = T \setminus Y$;
(iii) $F' = F \setminus Int(Y)$ onde $Int(Y)$ é o conjunto dos arcos internos de Y.

Elementos próprios

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).

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.

A page may contain one ore more *substitution transitions*.

- Each substitution transition is related to a *page*, i.e., a *subnet* providing a *more detailed description* than the transition itself.
- The page is a *subpage* of the substitution transition.

There is a *well-defined interface* between a substitution transition and its subpage:

- The places surrounding the substitution transition are *socket places*.
- The subpage contains a number of *port places*.
- Socket places are *related* to port places – in a similar way as actual parameters are related to formal parameters in a procedure call.
- A socket place has always the *same marking* as the related port place. The two places are just *different views* of the same place.

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

Estamos a um mês do final do curso e isso implica em:

1. acelerar o artigo final;
2. subir o nível dos exercícios de modelagem e análise
3. tornar estes exercícios mais próximos de uma modelagem real, o que significa descolar da "lista de exercícios" convencional;
4. maior disciplina no acompanhamento do curso.

Apresentação do artigo final

Fim