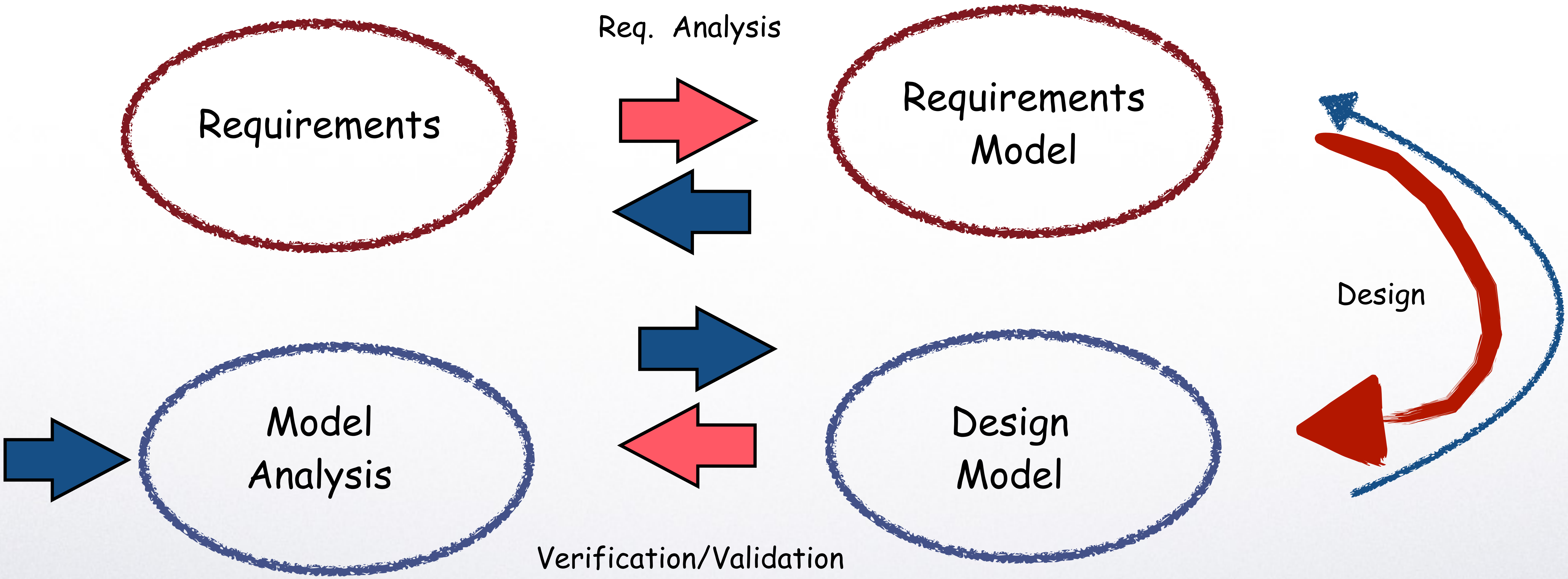
PMR 5237
Modelagem e Design de Sistemas Discretos em Redes de Petri

Aula 13 : Modelagem em Redes de Petri - Novas Perspectivas

Prof. José Reinaldo Silva
reinaldo@usp.br

Use of Petri Nets in Design



PN Basic Properties

1) *boundedness*, characterising finiteness of the state space.
2) *liveness*, related to potential fireability in all reachable markings. *Deadlock-freeness* is a weaker condition in which global infinite activity (i.e. fireability) of the net system model is guaranteed, but some parts of it may not work at all.
3) *reversibility*, characterising recoverability of the initial marking from any reachable marking.
4) *mutual exclusion*, dealing with the impossibility of simultaneous *submarkings* (p-mutex) or *firing concurrency* (t-mutex).

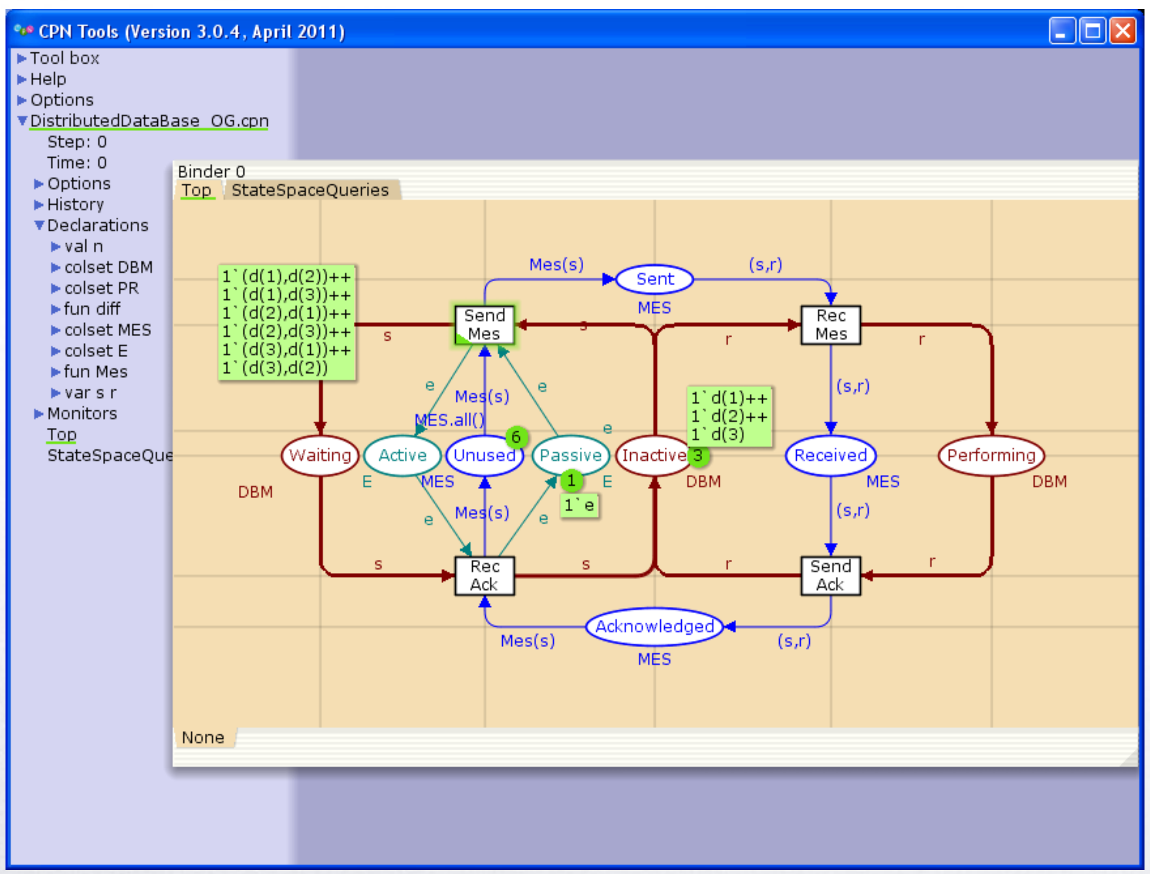
Fairness

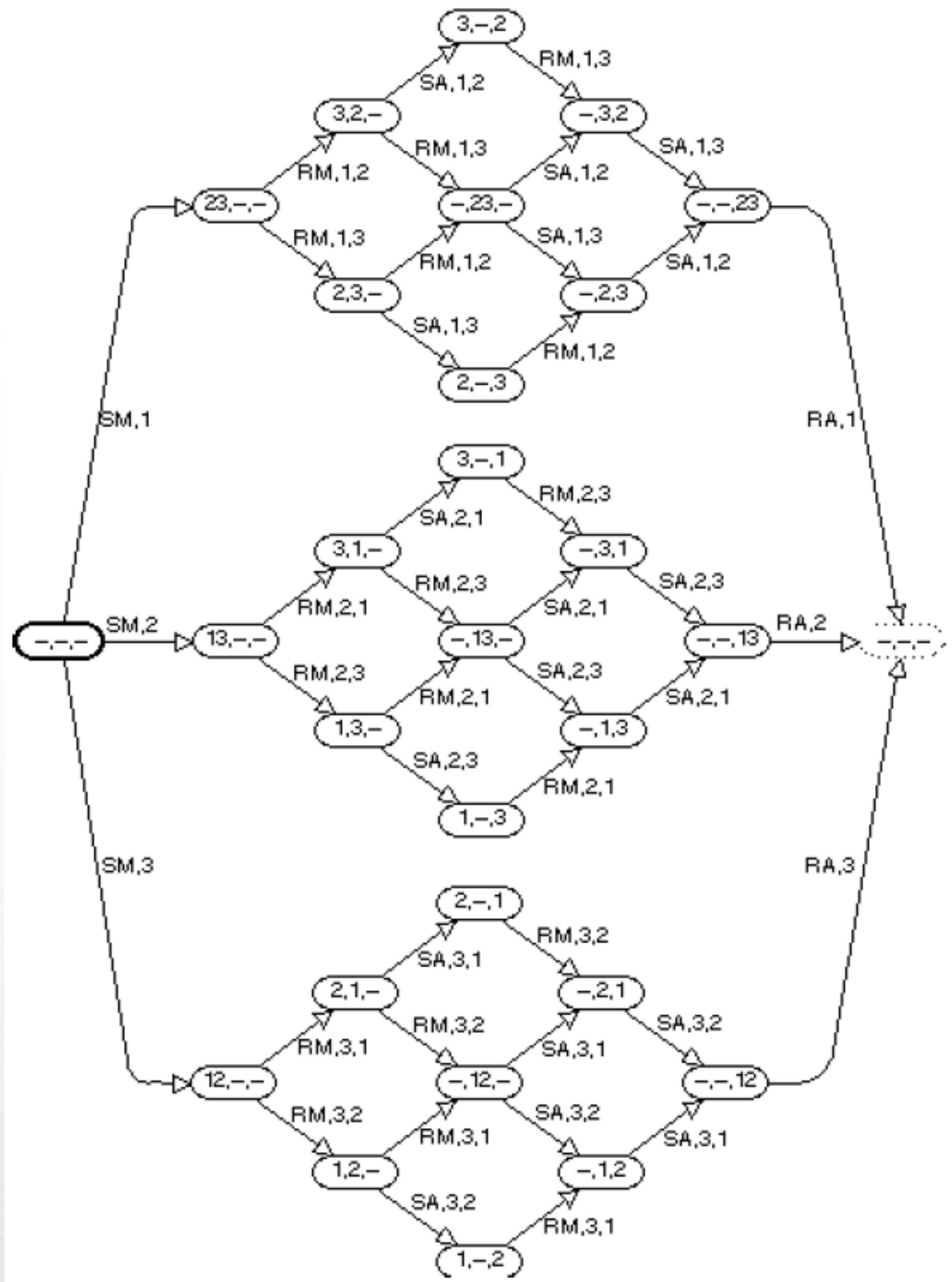
Invariants

Going Further with Discrete Events Modeling

- Analysing a larger set of properties;
- Scaling the analysis process (hierarchies, etc.);
- Improve abstraction and theoretical level;
- Providing more sophisticated support environments.

Analysis in CPN Nets





Theoretical and practical reasons to use CPN Nets

Improve abstraction (functional programming)

Improve the formal semantics of PN

Sound execution based sound support tools

Higher-Order and Symbolic Computation, 17, 207–243, 2004
© 2004 Kluwer Academic Publishers. Manufactured in The Netherlands.

Implementing Coloured Petri Nets Using a Functional Programming Language

LARS MICHAEL KRISTENSEN* lmkristensen@daimi.au.dk
SØREN CHRISTENSEN schristensen@daimi.au.dk
Department of Computer Science, University of Aarhus, Aabogade 34, DK-8200 Aarhus N, Denmark

Abstract. Coloured Petri Nets (CPNs) are a graphically oriented modelling language for concurrent systems based on Petri Nets and the functional programming language Standard ML. Petri Nets provide the primitives for modelling concurrency and synchronisation. Standard ML provides the primitives for modelling data manipulation and for creating compact and parameterisable CPN models.

Functional programming and Standard ML have played a major role in the development of CPNs and the CPN computer tools supporting modelling, simulation, verification, and performance analysis of concurrent systems. At the modelling language level, Standard ML has extended Petri Nets with the practical expressiveness required for modelling systems of the size and complexity found in typical industrial projects. At the implementation level, Standard ML has been used to implement the formal semantics of CPNs that provide the theoretical foundation of the CPN computer tools.

This paper provides an overview of how functional programming and Standard ML are applied in the CPN modelling language and the supporting computer tools. We give a detailed presentation of the key algorithms and techniques used for implementing the formal semantics of CPNs, and we survey a number of case studies where CPNs have been used for the design and analysis of systems. We also demonstrate how the use of a Standard ML programming environment has allowed Petri Nets to be used for the implementation of systems.

Keywords: distributed and concurrent computation, implementation techniques, programming environments and tools, Coloured Petri Nets, high-level Petri Nets, Petri Nets

1. Introduction

An increasing number of system development projects are concerned with concurrent systems. Examples of these range from large scale systems in the areas of telecommunication and applications based on WWW technology, to medium or small scale systems, in the area of embedded systems. The development of such systems is complex, a main reason being that the execution of a concurrent system consisting of a number of independent but co-operating processes may proceed in many different ways, e.g., depending on messages lost, the speed of the processes involved, and the time at which input is received

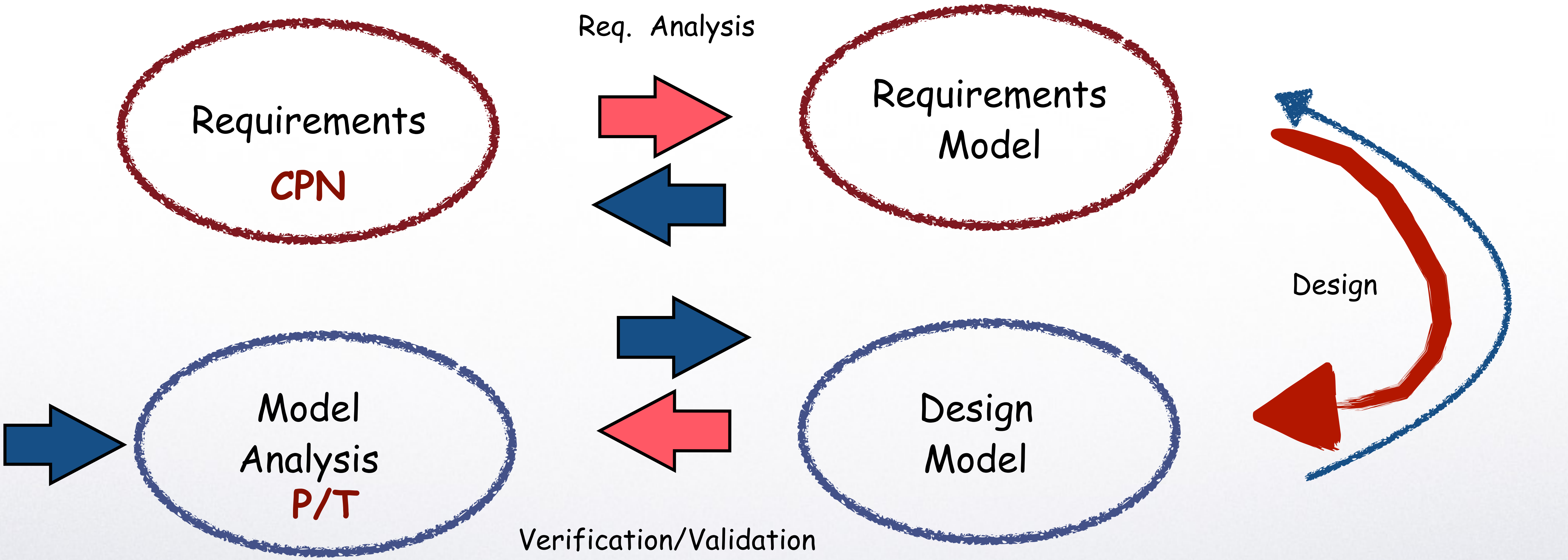
What are the drawbacks?

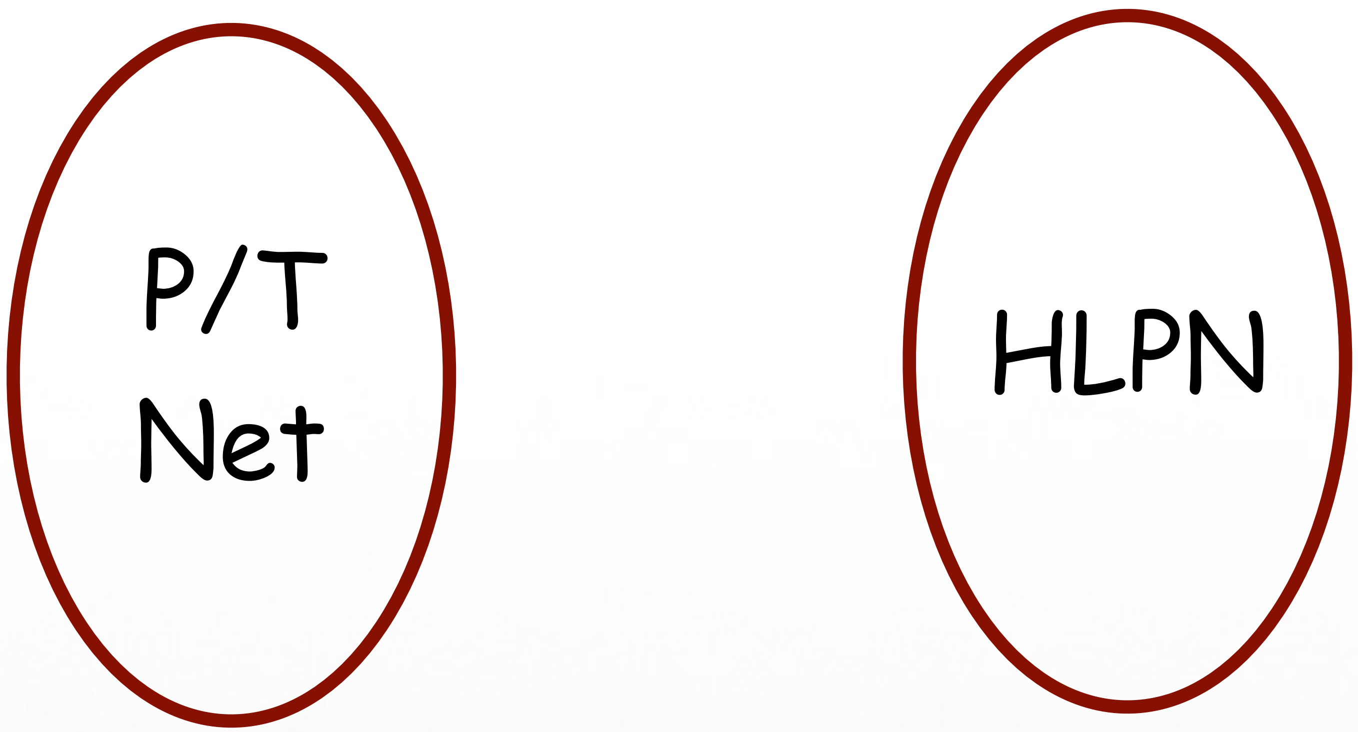
CPN was designed to improve semantic analysis but not to fit together with classic PN.

Automation of the analysis of some structural and behavioural properties was lost.

Improve the design approach but detach from control and direct implementation.

Use of Petri Nets in Design





That implies in providing a better matching between classic and high level methods.

- **15909-1:** Estabelece uma **definição única** de rede Place/Transition (P/T) e de rede de alto nível (HLPN) [ISO/IEC 2004]
- **15909-2:** Define um **formato de transferência** para o intercâmbio de modelos em RdPs (PNML) [ISO/IEC 2011]
- **15909-3:** Será dedicada à padronização das **extensões** das RdP, incluindo a hierarquia, o tempo e as funcionalidades estocásticas [Hillah et al. 2006]

Online Browsing Platform (OBP)

Search | ISO/IEC 15909-1:2019(en) ×

ISO/IEC 15909-1:2019(en) Systems and software engineering — High-level Petri nets — Part 1: Concepts, definitions and graphical notation

Table of contents

Redlines ▼

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

This second edition cancels and replaces the first edition (ISO/IEC 15909-1:2004), which has been technically revised.

The main change compared to the previous edition is as follows:

— a complete redrafting of the concepts and definitions of Petri nets and Petri net types in a simpler, modular and incremental way.

A list of all parts in the ISO/IEC 15909 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Basic concepts of HLPN

An HLPN net will be defined by...

- a net graph NG that defines its syntax,
- an underlying algebraic specification that defines the semantic domain,
- a net inscription mapping syntactic entities to their semantic denotation

An HLPN is a structure:

$$N = (NG, \text{Sig}, V, H, \text{Type}, AN, M_0).$$

- $NG = (P, T; F)$ is a net graph, with:
  - P a finite set of nodes, called places;
  - T a finite set of nodes, called transitions, disjoint from $P (P \cap T = \varnothing)$; and
  - $F \subseteq (P \times T) \cup (T \times P)$ a set of directed edges called arcs, known as the flow relation.
- $Sig = (S, O)$ is a Boolean signature, where S is a set of sorts and where O is a set of operators defined in the Annex A of ISO/JEC (2002).
- V is an S-indexed set of variables, disjoint from O.
- $H = (S_H, O_H)$ is a many-sorted algebra for the signature Sig, defined in this list:
- Type: $P \rightarrow S_H$ is a function that assigns types to places.
- $AN = (A, TC)$ is a pair of net annotations.
  - A: $F \rightarrow \text{TERM}(O \cup V)$ such that for all $(p, t), (t, p) \in F$ and all bindings α, $Val_\alpha(A(p, t))$, $Val_\alpha(A(t', p)) \in \mu\text{Type}(p)$. TERM $(O \cup V)$, α, Val_α and $\mu\text{Type}(p)$ are defined in Annex A of ISO/IEC (2002). A is a function that annotates each arc with a term that when evaluated (for some binding) results in a multiset over the associated place's type.
  - TC: $T \rightarrow TERM(O \cup V)_{\text{Bool}}$ is a function that annotates transitions with Boolean expressions.
- $M_0 : P \rightarrow \cup_{p \in P} \mu$ Type(p) such that $\forall p \in P$, $M_0(p) \in \mu$ Type (p) is the initial marking function that associates a multiset of tokens (of the correct type) with each place.

Definition 4.3.1. *A* coloured Petri net *(CPN) is defined by a tuple* $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post}, \mathcal{C}, cd \rangle$, *where*

- P *is a finite set (the* set of places *of* $\mathcal{N}$*),*
- T *is a finite set (the* set of transitions *of* $\mathcal{N}$*), disjoint from* P,
- $\mathcal{C}$ *is the set of* colour classes,
- $cd\colon P \cup T \rightarrow \mathcal{C}$ *is the* colour domain mapping, *and*
- $\mathbf{Pre}, \mathbf{Post} \in \mathcal{B}^{|P| \times |T|}$ *are matrices (the* backward *and* forward incidence matrices *of* $\mathcal{N}$*) such that* $\mathbf{Pre}[p,t] : cd(t) \rightarrow \mathrm{Bag}(cd(p))$ *and* $\mathbf{Post}[p,t] : cd(t) \rightarrow \mathrm{Bag}(cd(p))$ *are mappings for each pair* $(p,t) \in P \times T$.

$\mathcal{B}$ *can be taken as the set of mappings of the form* $f : cd(t) \rightarrow \mathrm{Bag}(cd(p))$. *Again,* $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$ *is called the* incidence matrix.

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)) = $C(p)_{MS}$].

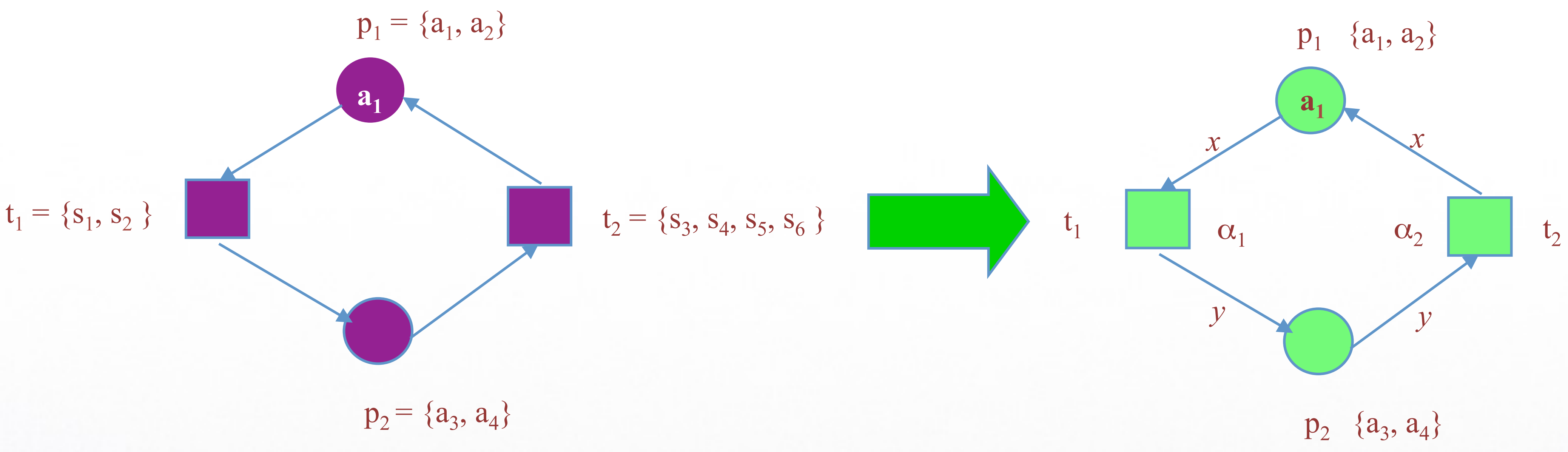
A marking in a HLPN net is a multiset defined by

$$M: P \to \cup_{p \in P} \mu \ \text{Type}(p) \quad \text{such that} \quad \forall p \in P, M(p) \in \mu \ \text{Type}(p).$$

Definition 35

Seja uma rede P/T com estrutura N, $PT = (N, K, W, M_0)$. e uma bijeção (equivalência) ρ que preserva o sort. Chama-se rede quociente em relação a ρ ao sistema $\bar{PT} = (\bar{N}, \bar{K}, \bar{W}, \bar{M_0})$ que tem a mesma dinâmica que a rede original.

Basic High Level Net



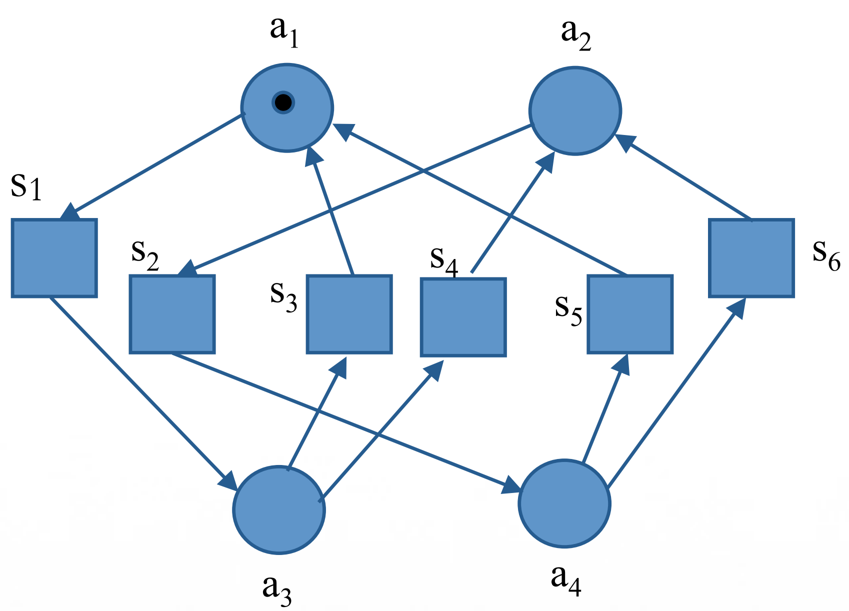
- lugares possuem marcas individualizadas
- transições ocorrem em diferentes modos restritos à regra de localidade (agir somente sobre o seu pre-set e pós-set)

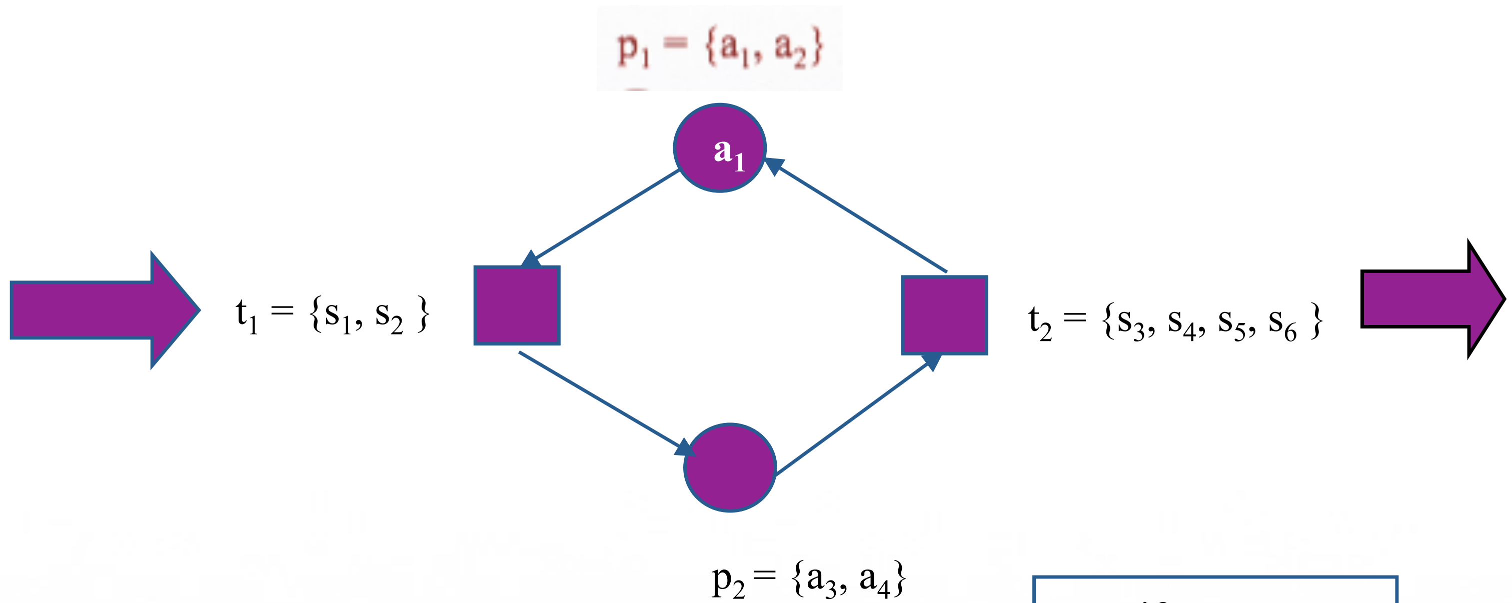
α_1 = if $x = a_1$
then $y = a_3$
else if $x = a_2$
then $y = a_4$

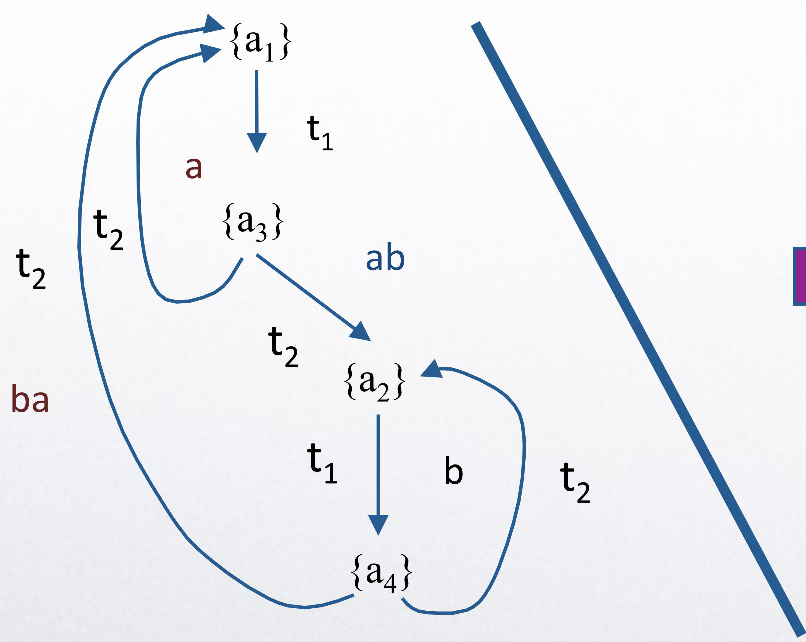
α_2 = if $y = a_3$
then $(x = a_1) \vee (x = a_2)$
else if $y = a_4$
then $(x = a_1) \vee (x = a_2)$

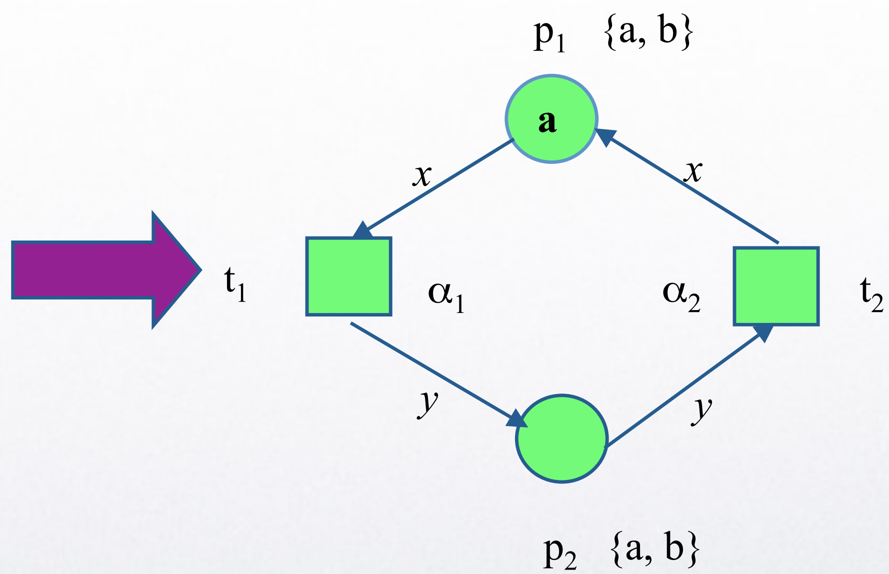
$P = \{a_1, a_2, a_3, a_4\}$

Smith, E.; Principles of High Level Nets, LNCS, 1491, Springer Verlag, 1998.









α_1 = if x= a
then y = a
else if x= b
then y =b

α_2 = if y= a
then (x = a) ∨ (x = b)
else if y= b
then (x = a) ∨ (x = b)

$P = \{a, b_2\}$

Formalmente, cada rede P/T está relacionada com uma rede quociente…

Definition 34

Seja uma rede place/transition N e seja o seu domínio $X = P \cup T$. Existe uma equivalência ρ entre a rede N e sua rede quociente $\bar{N}$ tal que:

i) $\forall x \in X, \exists \bar{x}$ que denota uma classe de equivalênica $\{y \in X \,|\, x \,\rho\, y\}$.

ii) Seja $Y \subseteq X$, então $\bar{Y} := \{\bar{x} \,|\, x \in Y\}$,

iii) ρ preserva o sort, isto é, $\rho \cap (P \times T) = \emptyset$,

iv) A relação de fluxo $\bar{F}$ sobre o dominio $\bar{X}$ é definida por,

$$\bar{x} F \bar{y} \iff \exists x\prime \in \bar{x} \wedge \exists y\prime \in \bar{y} \,|\, x \, F y$$

v) Denota-se a nova rede $(\bar{P}, \bar{T}; \bar{F})$ de $\bar{N}$.

Veremos mais adiante um processo chamado unfolding (“desdobramento”), que consiste justamente em desfazer os dobramentos resultantes do processo de fatoração.

Trata-se portanto em desfazer o colapso das simetrias que levaram aos dobramentos, e à identificação das marcas nas redes de alto nível/coloridas. Portanto, no unfolding cada tipo de marca deve ter sua própria sub-rede, simétrica e funcionalmente equivalentes entre si. A rede resultante deste processo é novamente uma rede P/T.

Dada uma rede P/T é sempre possível dobrá-la e obter com isso a correspondente rede quociente. Por sua vez a rede quociente pode ser trabalhada (baseada em sorts e teoria de tipos) para obter a rede de alto nível equivalente. É possível também traçar o caminho inverso e, à partir da rede de alto nível, obter a rede equivalente P/T, usando para isso as técnicas de unfolding já descritas.

Definition 4.11. Let $\overline{\text{PT}} = (\bar{N}, \bar{K}, \bar{W}, \bar{M}_{\text{in}})$ be a quotient system. The basic high level net $(\bar{N}, D, \Phi, \tilde{W}, \tilde{\bar{K}}, \bar{M}_{\text{in}})$, where D and Φ are defined according to (4.10), is called the BHL *associated with* $\overline{\text{PT}}$.

Definition 4.13. The *unfolded system associated* with a basic high level net BHL is the PT-net $\widetilde{\text{BHL}} = (\tilde{P}, \tilde{T}, \tilde{F}, \tilde{W}, \tilde{K}, \tilde{M}_{\text{in}})$ as defined above.

It is immediately verified that a *strict* BHL gives rise to an *elementary net system*.

Proposition 4.14.

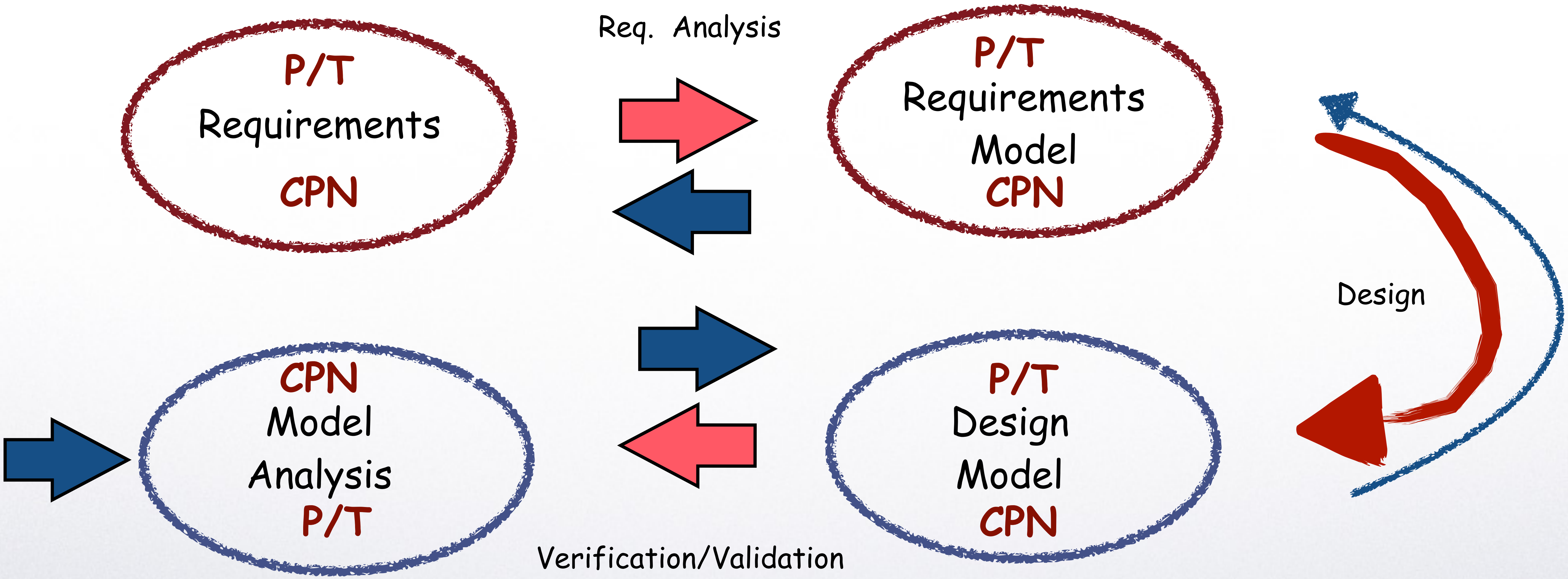
(i) *Let* $\overline{\mathrm{PT}}$ *be a quotient of a PT-net* PT. *Let* $\widetilde{\overline{\mathrm{PT}}}$ *be the unfolding of* $\overline{\mathrm{PT}}$ *according to Definition 4.13. Then, except for names of elements,* PT *and* $\widetilde{\overline{\mathrm{PT}}}$ *are identical. In particular, transition-enabling and -occurrence in both systems coincide.*

(ii) *Let* $\widetilde{\mathrm{BHL}}$ *be the PT-net obtained from a basic high-level net* BHL *according to Definition 4.13. Let* $\overline{\widetilde{\mathrm{BHL}}}$ *be the quotient of* $\widetilde{\mathrm{BHL}}$ *with respect to the equivalence* ρ *given by*

$$(x, y)\, \rho\, (x', y') :\Leftrightarrow x = x'.$$

Then the systems BHL *and* $\overline{\widetilde{\mathrm{BHL}}}$ *are identical, except for names of elements. In particular, transition-enabling and -occurrence coincide.*

Use of Petri Nets in Design



d-l@b

MECATRÔNICA

Principles of High-Level Net Theory

Einar Smith

GMD, Sankt Augustin, Germany, esmith@gmd.de

Abstract. The paper gives an introduction to fundamentals and recent trends in the theory of high-level nets. High-level nets are first formally derived from low-level nets by means of a quotient construction. Based on a linear-algebraic representations, we develop an invariant calculus that essentially corresponds to the algebraic core of the well-known coloured nets. We demonstrate that the modelling power of high-level nets stems from the use of expressive symbolic annotation languages, where as a typical model we consider predicate-transition nets, both concrete models and net-schemes. As examples of specific high-level analysis-tools we discuss symbolic place-invariants and reachability-trees

- **15909-1:** Estabelece uma **definição única** de rede Place/Transition (P/T) e de rede de alto nível (HLPN) [ISO/IEC 2004]

Redes Simétricas

- **15909-2:** Define um **formato de transferência** para o intercâmbio de modelos em RdPs (PNML) [ISO/IEC 2011]

- **15909-3:** Será dedicada à padronização das **extensões** das RdP, incluindo a hierarquia, o tempo e as funcionalidades estocásticas [Hillah et al. 2006]

Redes Orientadas a Objetos

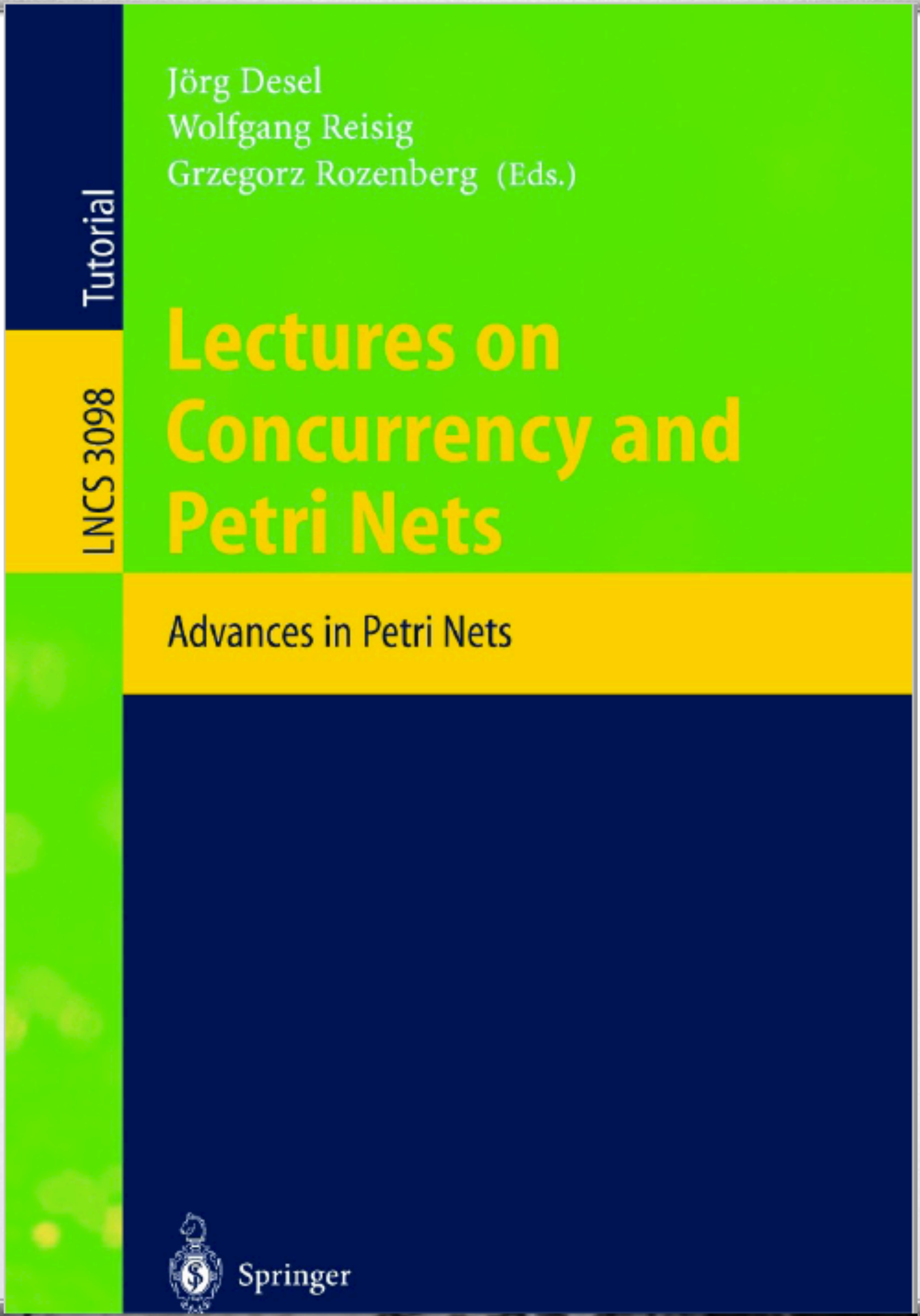
Object Petri Nets
Using the Nets-within-Nets Paradigm

Rüdiger Valk

Universität Hamburg, Vogt-Kölln-Str.30, D-22527 Hamburg, Germany
valk@informatik.uni-hamburg.de

Abstract. The nets-within-nets paradigm provides an innovative modelling technique by giving tokens themselves the structure of a Petri net. These nets, called *token nets* or *object nets*, also support the object oriented modelling technique as they may represent real world objects with a proper dynamical behaviour. Between object nets and the surrounding net, called *system net*, various interaction mechanisms exist as well as between different object nets. This introduction into the field of object Petri nets starts with small examples and proceeds by giving formal semantics. Some of the examples are modelled within the formalism of the Renew tool. Finally the differences between reference and two kinds of value semantics are discussed.

1 Nets within Nets


Tutorial
LNCS 3098
Jörg Desel
Wolfgang Reisig
Grzegorz Rozenberg (Eds.)
Lectures on Concurrency and Petri Nets
Advances in Petri Nets
Springer

Para além da estruturação

Redes clássicas Hierárquicas e de alto Nível → Redes Orientadas a Objetos

Vantagens de uma abordagem orientada a objetos

- um modelo simples de concorrência e sincronização.
- semântica operacional que suporte simulação e execução.
- técnicas de análises.
- abstração
- refinamento
- encapsulamento
- reutilização e compartilhamento.

Redes de Petri e objetos

Propostas

- Tratamento por objetos do processo de geração da rede
- Exploração da dualidade e da hierarquia (e eventualmente da estruturação) ➔ herança simples
- Manutenção ou redefinição das propriedades das redes

O dilema Petri Nets versus objetos

Petri Nets inside object

X

Objects inside Petri Nets

Se considermos agora as redes de alto nível, estas podem ser representadas por redes por objetos onde a marca é também um objeto ("tipado"), significando um objeto onde somente a parte estática está definida. Neste caso reproduzimos as redes de alto nível ou coloridas.

Entretanto, se incluímos no objeto marca um comportamento distinto temos agora o que Rudiger Valk chama de “nets inside nets”, ou uma representação de segunda ordem.

Let X be a generic “agent”,

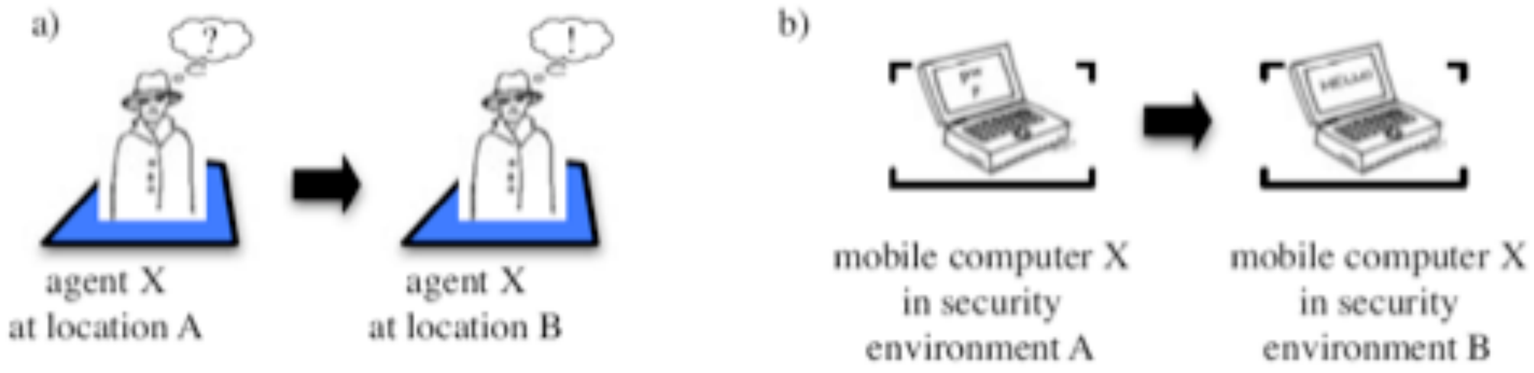


Fig. 1. Moving objects I

Valk, R., Object Petri Nets: Using Nets within Nets Paradigm, LNCS 3098, pp 819-848, J. Desel, W. Reisig, G. Rosenberg, (eds.), Spring Verlag, 2004.

Fusão Petri Nets & Objetos

Objects → PN

- **A classe mãe não pode existir sem sacrificar a dualidade das redes, que gera a maioria das propriedades usadas em análise**
- **É difícil achar um significado para o polimorfismo nas redes**
- **Não é possível (ou pelo menos é difícil) achar um uso para a herança múltipla**

No.	Attribute	Values
1	paradigm	state machine, algebra, process algebra, trace
2	formality	informal, semi-formal, formal
3	graphical representation	yes, no
4	object-oriented	yes, no
5	concurrency	yes, no
6	executability	yes, no
7	usage of variables	yes, no
8	non-determinism	yes, no
9	logic	yes, no
10	provability	yes, no
11	model checking	yes, no
12	event inhibition	yes, no

method name	paradigm	formality	graphical represen-tation	object-oriented
Action Systems	state transition	formal	no	no
B	state transition	formal	no	no
CASL	algebra	formal	no	yes
Cleanroom & JSD	traces & process algebra	formal	yes	no
COQ	state transition	formal	no	no
Estelle	state transition	formal	no	no
LOTOS	process algebra	formal	no	yes
OMT & B	state transition	formal	yes	yes
Petri Nets	state transition	formal	yes	no
Petri Nets with Objects	state transition	formal	yes	yes
SART	state transition	informal & semi-formal	yes	no
SAZ	state transition	semi-formal & formal	yes	no
SCCS	process alge-bra	formal	no	no
SDL	state transition	formal	yes	yes
UML	state transition	informal & semi-formal	yes	yes
VHDL	state transition	formal	no	no
Z	state transition	formal	no	no

method name	concurrency	executability	usage of variables	non-determinism
Action Systems	no	yes	yes	yes
B	no	yes	yes	yes
CASL	no	yes	yes	no
Cleanroom & JSD	no	yes	yes	yes
COQ	no	yes	yes	yes
Estelle	yes	yes	yes	no
LOTOS	yes	yes	yes	yes
OMT & B	no	yes	yes	yes
Petri Nets	yes	yes	no	yes
Petri Nets with Objects	yes	yes	yes	yes
SART	yes	no	no	yes
SAZ	no	yes	yes	yes
SCCS	yes	yes	yes	yes
SDL	yes	yes	no	yes
UML	yes	no	no	no
VHDL	yes	yes	yes	no
Z	no	yes	yes	yes

Fim