Conceptual Modeling of Concurrent Systems through Stepwise Abstraction Using Petri Net Morphisms

Boleslaw Mikolajczak & Zuyan Wang
Computer and Information Science Department
University of Massachusetts Dartmouth
bmikolajczak@umassd.edu

Computer Science Seminar Series
December 5, 2003

Contents

- Motivation
- Related Work
- Objectives
- Terminology of Petri Net Morphisms
- Relations Between Petri Net Morphisms
- Structural Properties of Petri Net Morphisms
- Behavioral Properties of Petri Net Morphisms
- Example of WfM: On-Line Shopping System
- Conclusions and Future Work
- References

Motivation

- **Design Features supported by Petri Nets**:
  - Modeling and *Formal Specification* with Visual Programming paradigm
  - Representation of *Causality, Choice*, and *Concurrency*
  - Model *Verification* and *Validation* Capabilities
  - *Model Execution* Capabilities
  - Support for *Hierarchical Constructs*: Hierarchical Substitution, Hierarchical Invocation, Place Fusion (for Colored PNs)
  - *Reduction Analysis &Transformations'* Preserving Properties
  - *Rapid Prototyping* of Parallel and Distributed Systems
  - *Time Annotations* and *Performance Evaluation*.

Motivation, ctnd.

- In the development of complex concurrent systems specified with Petri nets, **refinement and abstraction operations** that preserve certain structural and behavioral properties, are desired and useful concepts
- **net morphism is a homogenous formalism** to represent models of different layers (i.e. relations between models can be expressed in a uniform way)
- **examples of relations between models**: abstraction from details; composition of sub-components (with properties of sub-components deduced from respective properties of the whole system and properties of subparts carry over to the composed system)

Motivation, ctnd.

- PNs as methodology and technology of **coordination**
- PN morphisms as **vertical structuring technique** in software engineering - stepwise abstraction-based modifications of nets
- **local states** and **atomic actions** are crucial in distributed computing
- occurrence of an atomic action affects a subset of local states
- it is important to prove **decisive properties** of distributed algorithms: safety ("*nothing bad will ever happen*" and liveness ("*eventually something good will happen*")

Motivation, ctnd.

- **Concurrent system engineering** - continuous verification is needed during all phases of the software development process
- entirely new verification/validation at each step is usually considered to be too expensive and time consuming
- **reduction analysis** deals with the reduction of PNs by replacing subnets of the net with less complex subnets such that some *desired properties remain invariant*
- once sufficiently reduced Petri net is achieved, the analysis can be performed

Motivation, ctnd.

- Using PN morphisms as **vertical structuring technique** there is no need to do the verification of PN model at each level of the development process
- PN morphisms - an **abstraction technique**
- PN morphisms - a method of **hierarchical modeling**
- top-down software development - **refinement**
- bottom-up software development - **abstraction**

Related Work

- **Petri net-based Refinement Techniques:**
  - **Padberg -** Rule-based refinement (1998)
  - **Devillers** - General refinement (1997)
  - **Fehling** - Hierarchical modeling (1991).
- **Abstraction techniques (for P/T Petri nets):**
  - **Berthelot's reduction** - preserves liveness, safety, and boundedness (1987)
  - **Cheung's invariant preserving transformation** - preserves liveness, boundedness, and invariants (1998)
  - **Lee's general Petri net reduction** - preserves liveness and boundedness (1987).

Objectives

- **To examine and prove** the structural (i.e. syntactic description of a PN as a graph) and behavioral properties that Petri net morphisms preserve (morphisms are structure-preserving mappings; morphisms are basis of many operations on nets such as abstractions, composition, hierarchy, and foldings)
- **To study relationships** between vicinity respecting Petri net morphisms, Winskel's Petri net morphisms, and general Petri net morphisms
- **To show an example** of the Petri net morphisms' application in the development of WfM concurrent systems (conceptual modeling)
- **To discuss roles of Petri net morphisms** in the development of concurrent systems and in creation of software tools to support development of such systems (work in progress)

Homomorphisms of Complete Finite Automata

- **For DFA** state, state-input, and generalized homomorphisms can be defined and used in systems' reductions using related *quotient automata*
- **Sequential and Parallel Algorithms** to compute a set of all homomorphisms between two deterministic complete automata do exist (Mikolajczak - 1979, 1994)
- **Automata homomorphisms** are equivalent to **right congruences** (called SP partitions) on a set of states
- a set of all automata homomorphisms defines a set of all **serial or parallel decompositions**
- **'catalog's paradigm'** as an example of the homomorphism concept

Net morphisms, foldings, epimorphisms, epifoldings, net isomorphisms

Let $N_1=(P_1,T_1,F_1)$ and $N_2=(P_2,T_2,F_2)$ be two nets and φ: $X_1\text{-->}X_2$, a mapping where $X_i=P_i \cup T_i$. Furthermore, define the **at-relation** $A_i=(F_i \cup F_i^{-1}) \cap (P_i x T_i)$, where i=1,2.

a) ϕ is said to be a **net morphism** if

1. for every $x,y \varepsilon P_1 \cup T_1$ $(x,y) \varepsilon F_1 => (\varphi(x), \varphi(y)) \varepsilon F_2$ or $\varphi(x)= \varphi(y)$, i.e. *preservation of F - flow relation*

2. for every $x,y \varepsilon P_1 \cup T_1$ $(x,y) \varepsilon A_1 => (\varphi(x), \varphi(y)) \varepsilon A_2$ or $\varphi(x)= \varphi(y)$, i.e. *preservation of A relation*, called "*continuity*"

b) a net morphism φ is a **folding** if $\varphi(P_1) \subseteq P_2$ and $\varphi(T_1) \subseteq T_2$

c) a net morphism φ is an **epimorphism** if φ is *surjective* and for every $(x_2,y_2) \varepsilon F_2$ there is an arc $(x_1,y_1) \varepsilon F_1$ such that $(x_2,y_2)=(\varphi(x_1), \varphi(y_1))$. A folding which is an epimorphism is called and **epifolding**.

d) a net morphism φ is a **net isomorphism** if φ is a *bijection* and φ^{-1} is also a net morphism.

Terminology of Net Morphisms

	Structure respecting PN	Behavior respecting PN
Places	P	P
Transitions	T	T
Flow relation	$F \subseteq (P \times T) \cup (T \times P)$	F is multiset of $(P \times T) \cup (T \times P)$
Initial marking	—	M_0 is a vector of multisets

Terminology of Net Morphisms, ctnd.

- **Petri net morphism** maps a structure of one net to another so as *to maintain structure and connectivity*, in the sense that arcs are maintained wrt directions and transition input/output character or else the endpoints are superposed
- Under what circumstances would a net morphism be called a **homomorphism**? Or, put it another way, a homomorphism preserves the results of applying operations - *which operators are relevant in net morphism?*
- What terminology should be applied to a morphism where the mapping does not just specify the transformation of the net structure but also of the **dynamic behavior** - the *markings*, *steps*, and *enabling of transitions* ?

Definition of a Net Morphism

- Let N=(P,T,F) and N′=(P′,T′,F′) be nets with adjacency relation of a net $A=(FUF^{-1})\cap(PxT)$. A mapping f: X --->X′ is called **net morphism**, denoted by f: N --->N′ if for every a,b in X the following properties hold:

  1. $(a,b)\varepsilon A \Rightarrow (f(a),f(b))\varepsilon A' \cup id_{X'}$
  2. $(a,b)\varepsilon F \Rightarrow (f(a),f(b))\varepsilon F' \cup id_{X'}$

  where, $id_{X'}=\{(x,x): x\varepsilon X'\}$
- **Theorem.** Let f: N --->N′ be a **net morphism**; then

  1. For every t and every p, $f(t)=p \Rightarrow f({}^{\bullet}t \cup \{t\} \cup t^{\bullet})=\{p\}$
  2. For every p and every t, $f(p)=t \Rightarrow f({}^{\bullet}p \cup \{p\} \cup p^{\bullet})=\{t\}$

Terminology of Net Morphisms, ctnd.

- **Transformations or generations of nets** (only source net and some information about the mapping is given; target net is given only implicitly; *target net is generated by the net morphism*)
- Given the partition of the elements which is generated by the equivalence relation '*is mapped to the same element*', the generated target net is unique up to isomorphism; these net morphisms are called **quotients**
- a **surjective net morphism** f: N --->N' is called **quotient** if for every $(x,y) \varepsilon F'$ there exists $(a,b) \varepsilon F$ with $f(a)=x$ and $f(b)=y$

Net Morphisms (NM)

- There are **two types** of Petri net morphisms: structural and behavioral Petri net morphisms
- **Structural Petri net morphisms** respect bipartition and the flow relation of the source net
- **Behavioral Petri net morphisms** focus on relationship between markings caused by firing of different transitions in both nets
- **Net morphisms** are structural PN morphisms; a net morphism is a mapping from elements of a source net to elements of a target net that respects the bipartition and flow relation of the source net

Vicinity Respecting Morphisms (VRM)

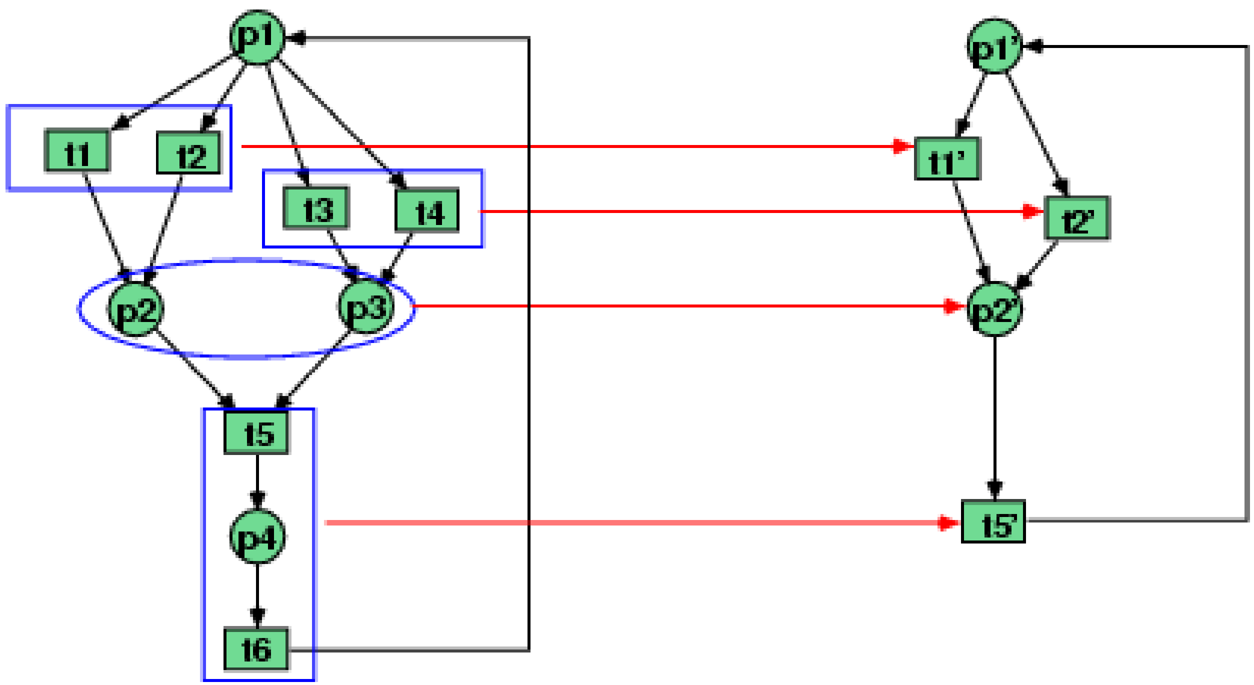
- **P-vicinity is preserved**, i.e. that pre/post-vicinity of a **place x** is mapped to either the pre/post-vicinity of the image of place x or the image of place x itself
- **T-vicinity is preserved**, i.e. the pre/post-vicinity of a **transition x** is mapped to either the pre/post-vicinity of the image of transition x or the image of transition x itself
- **VRMs** are restricted to a class of net morphisms
- for **VRMs** if two elements in PN are close enough to be mapped to the same element, then the effects of such mapping on the PN environment should be local

Multisets and Multirelations

- μ**X** - the set of **multisets** (**bags**) over X; it also means a space of multisets over X with X as a basis; μ**X** is a vector of values of function f: X -->N
- α**: X -->$_\mu$Y** - a **multi-relation** from X to Y, i.e. a matrix from X to Y with integer values of α(x,y)

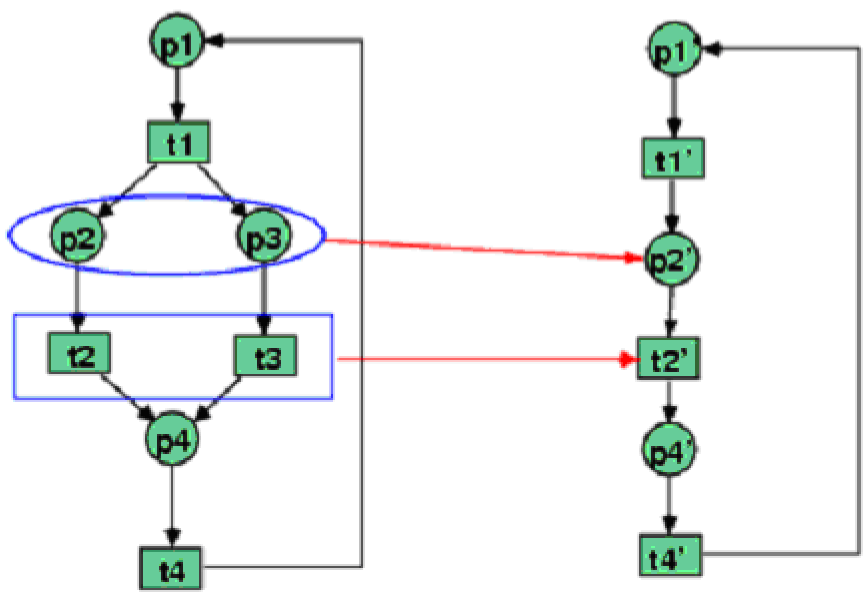
Net Morphisms (NM) - example

- Structural PN morphisms respect **bipartition** and **flow relation** of Petri nets:



Vicinity Respecting Morphisms (VRM) - example

- **Every VRM is a NM**, but not vice versa.



Winskel's Morphisms (WM)

	Homomorphisms	Morphisms
β	Multirelation $P \to_{\mu} P'$	Multirelation $P \to_{\mu} P'$
η	Multirelation $T \to_{\mu} T'$	Partial function $T \to_{\mu} T'$
Satisfying properties	$\beta M_0 = M_0^{'}; \quad \forall A \in \mu T,$ $^{\bullet}(\eta A) = \beta(^{\bullet}A) \,\&\, (\eta A)^{\bullet} = \beta(A^{\bullet})$	$\eta_{t,t'} \leq 1;$ $\eta_{t,t'} = 1 \,\&\, \eta_{t,t''} = 1 \Rightarrow t' = t''$

General Morphisms (GM)

A *general morphism* from $N=(P,T,F,M_0,\mathbf{M})$ to $N'=(P',T',F',M_0^{'},\mathbf{M'})$ is a partial function $\eta: T \to T'$ and a multirelation $\beta: P \to P'$ satisfying the following conditions:

(1). $\beta M_0 = M_0^{'}$

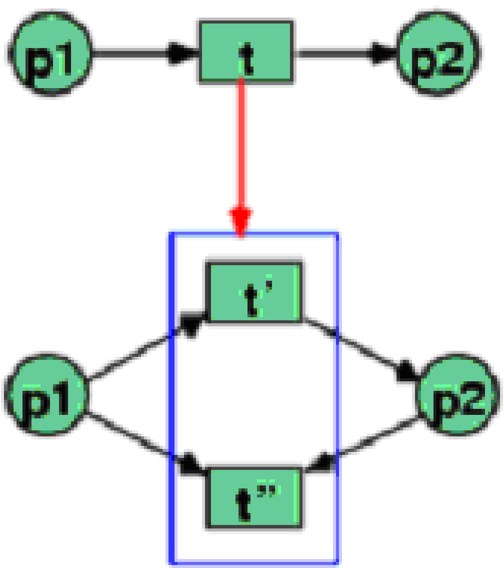
(2). $M \in \mathbf{M}$ implies $\beta M \in \mathbf{M'}$

(3). ${}^{\bullet}(\eta t) \le \beta({}^{\bullet}t)$

(4). $\beta(t^{\bullet}) = \beta({}^{\bullet}t) - {}^{\bullet}(\eta t) + (\eta t)^{\bullet}$.

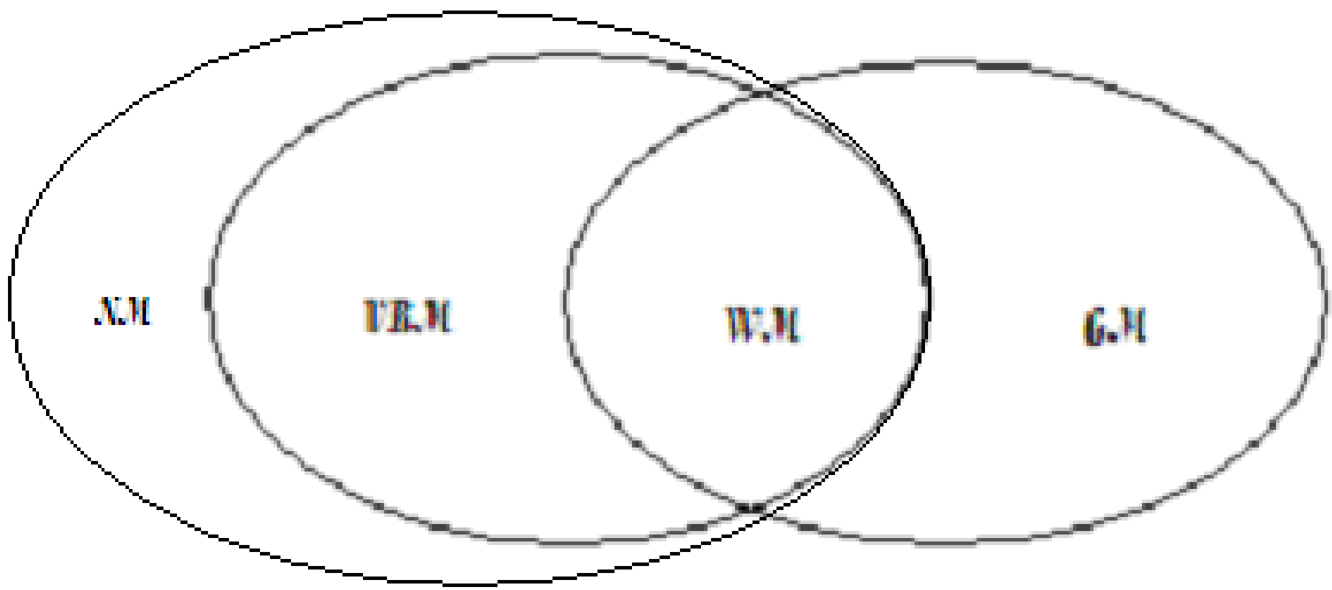
Relations Between NM, VRM, WM and GM - example

- **Every WM is a VRM**; but not vice versa



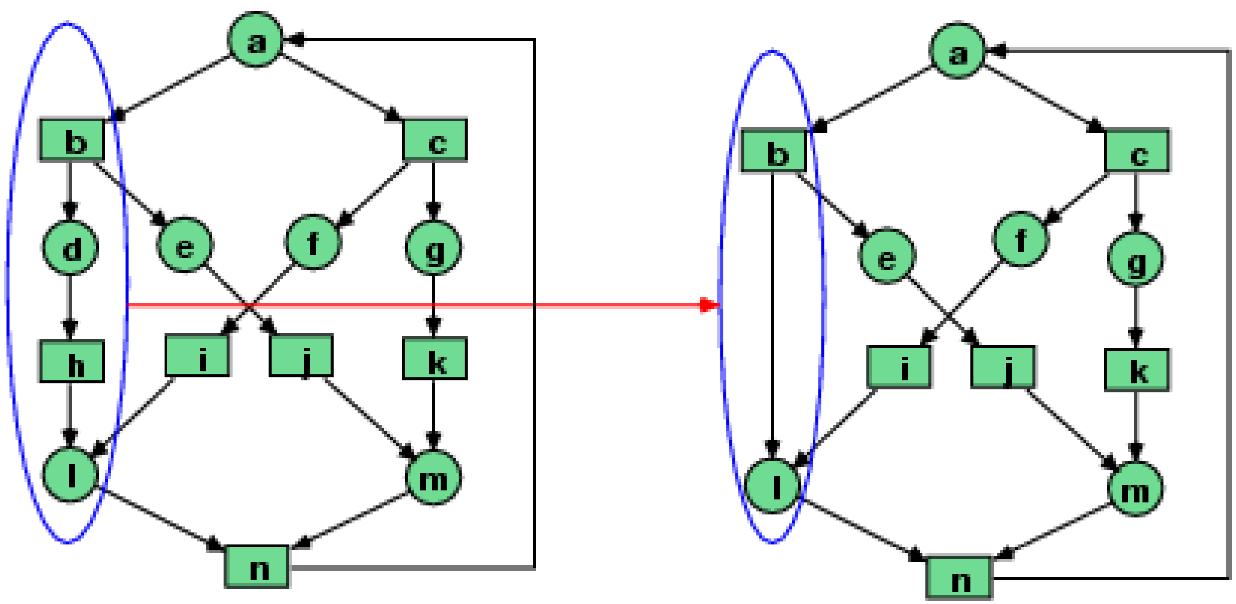
Relations Between NM, VRM, WM and GM, ctnd.

- **Every WM is a GM** but not vice versa



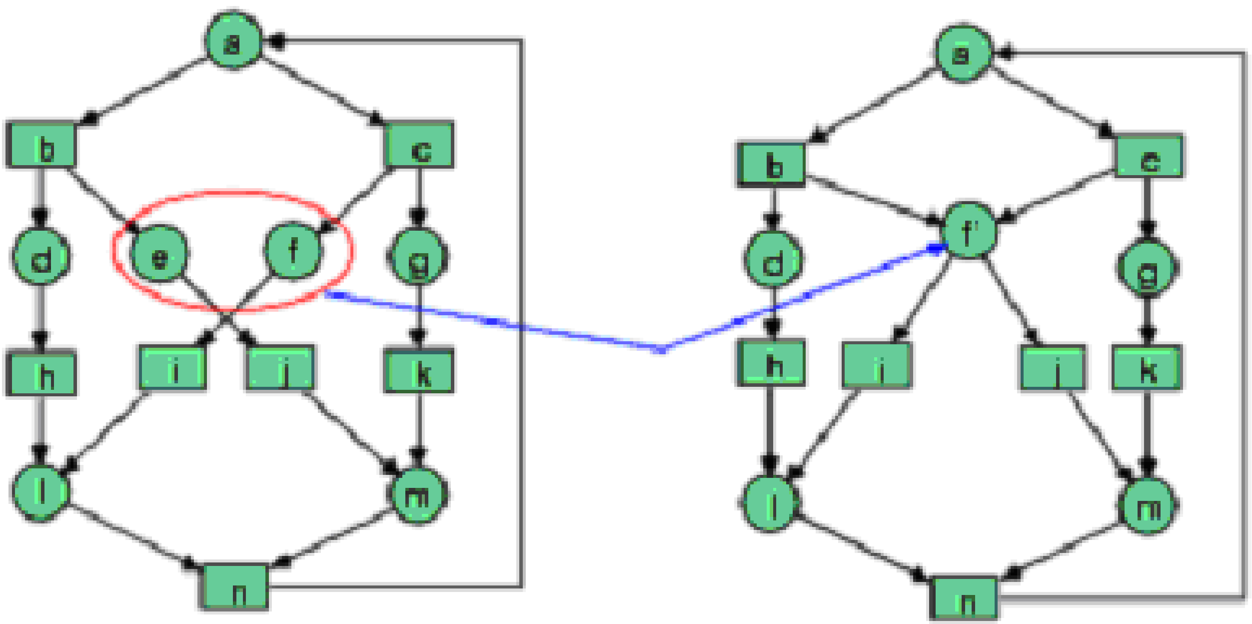
Structural Properties of Vicinity Respecting Morphisms

- **VRM** preserve **connectivity** of Petri nets
  - **VRM do not preserve edge** of Petri nets
  - **VRM do preserve path** in Petri nets



Structural Properties of Vicinity Respecting Morphisms, ctnd.

- **VRM** preserve **strong connectedness** of Petri nets

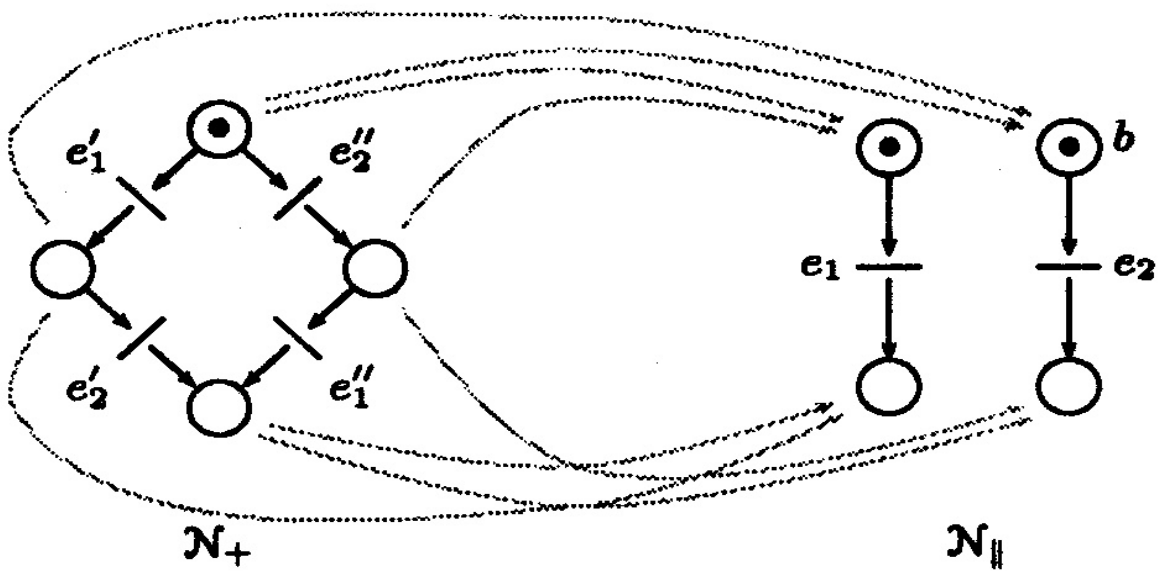


Structural Properties of Petri Net Morphisms, ctnd.

- **VRM preserve pre/post vicinity** of places/transitions in Petri nets
- **VRM preserve subnets** of Petri nets
- **VRM do not preserve place/transition bordered subnets** of Petri nets
- **WM preserve all the structural properties** that **VRM** preserve

Structural Properties of Petri Net Morphisms, ctnd.

- **GM do not preserve any structural properties** of Petri nets

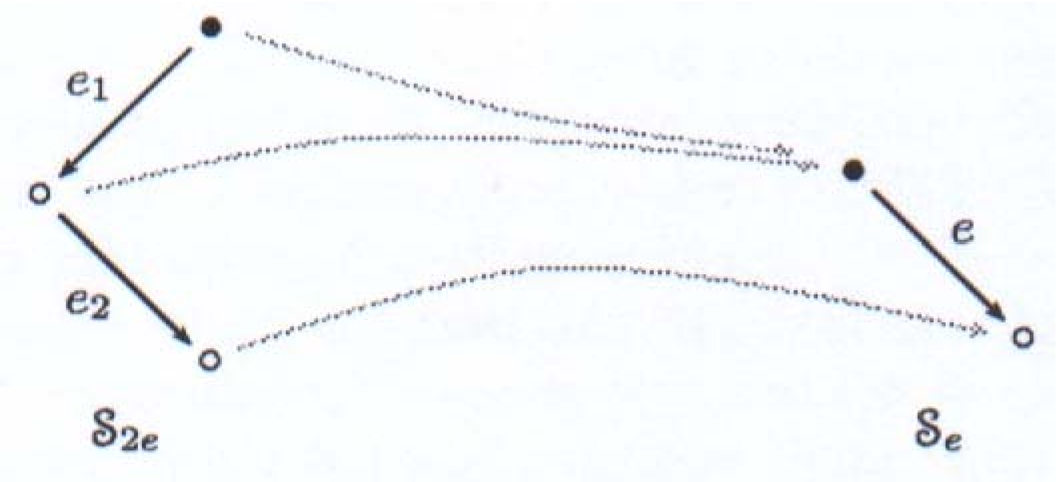


Behavioral Properties of Petri Net Morphisms - Discovery Matrix

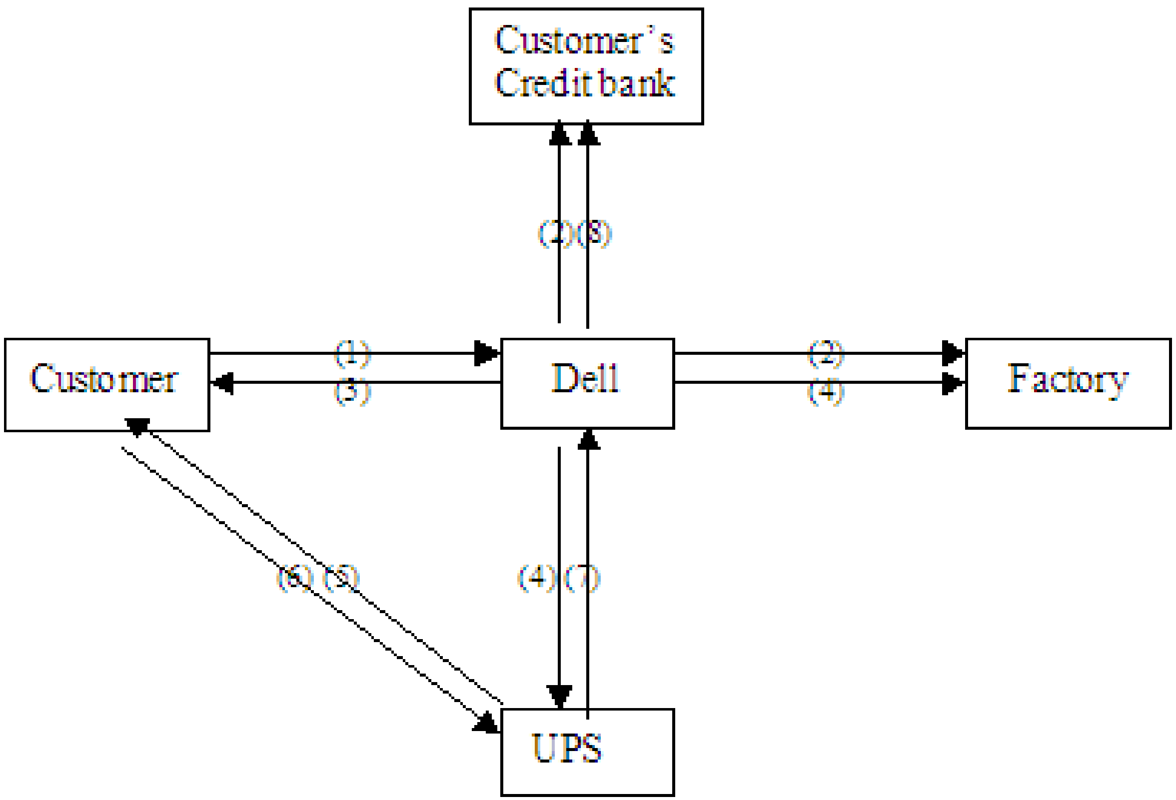
Morphisms / Behavioral properties	Winskel's morphisms (P/T Petri nets)	General morphisms (Augmented Petri nets)
Liveness	Yes (Pr. 4.16)	Yes (Pr. 4.17)
Boundedness	Yes (Pr. 4.12)	Yes (Pr. 4.13)
Safety	Yes (Pr. 4.14)	Yes (pr. 4.15)
Initial marking	Yes (Pr. 4.8)	Yes (Pr. 4.9)
Reachable marking	Yes (Pr. 4.10)	Yes (Pr.4.11)
Case graph	Yes (p.29)	Proved [BB99 Co.1]

Behavioral Properties of Petri Net Morphisms, ctnd.

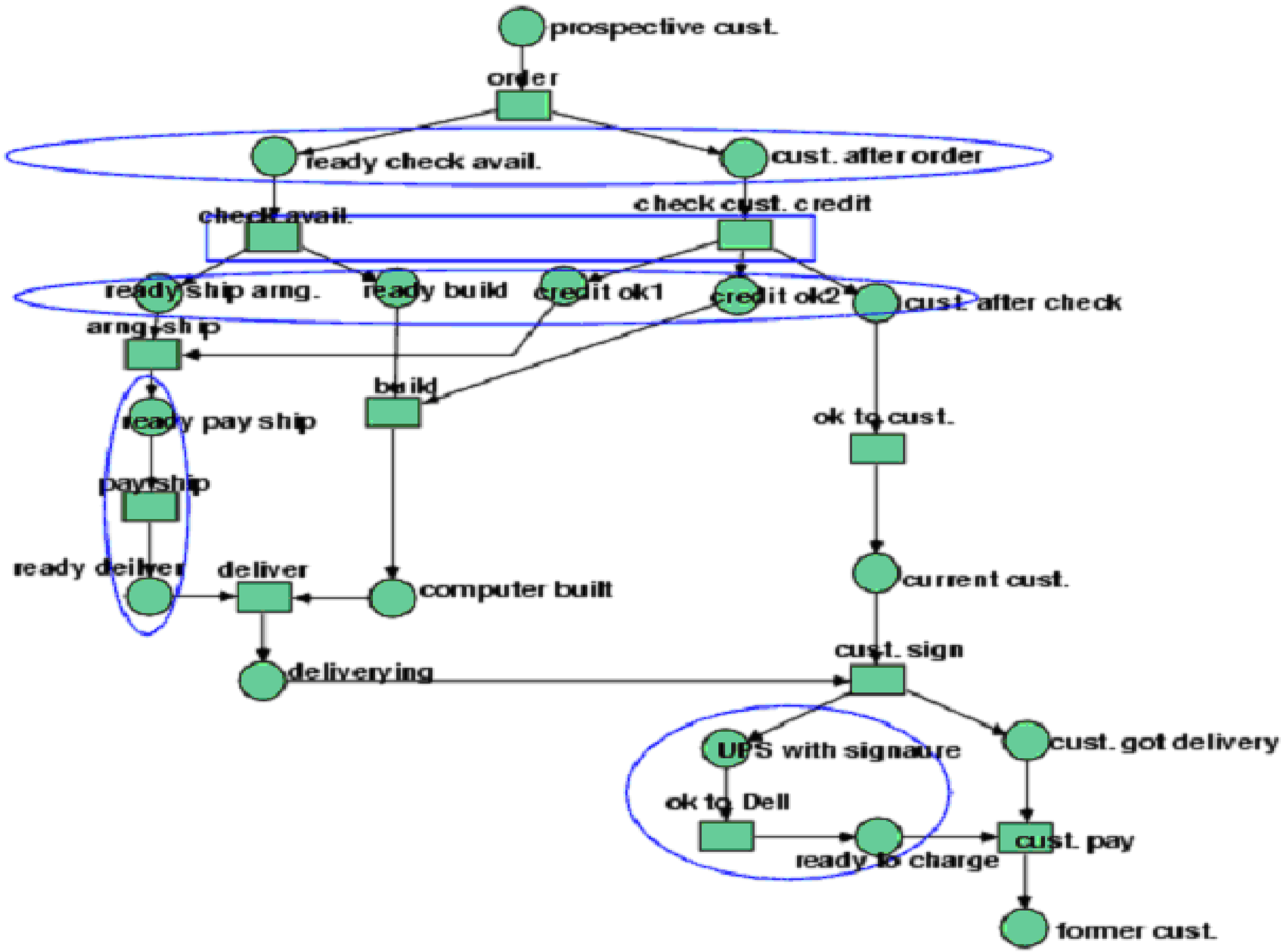
- **Case graph** describes global behavior of C/E Petri nets
- Given a GM/WM $(\eta, \beta): N \to N'$, (η, β) is also a morphism between the **case graphs** generated by N and N'. This morphism is coincident to the morphism definition for deterministic complete finite automata.



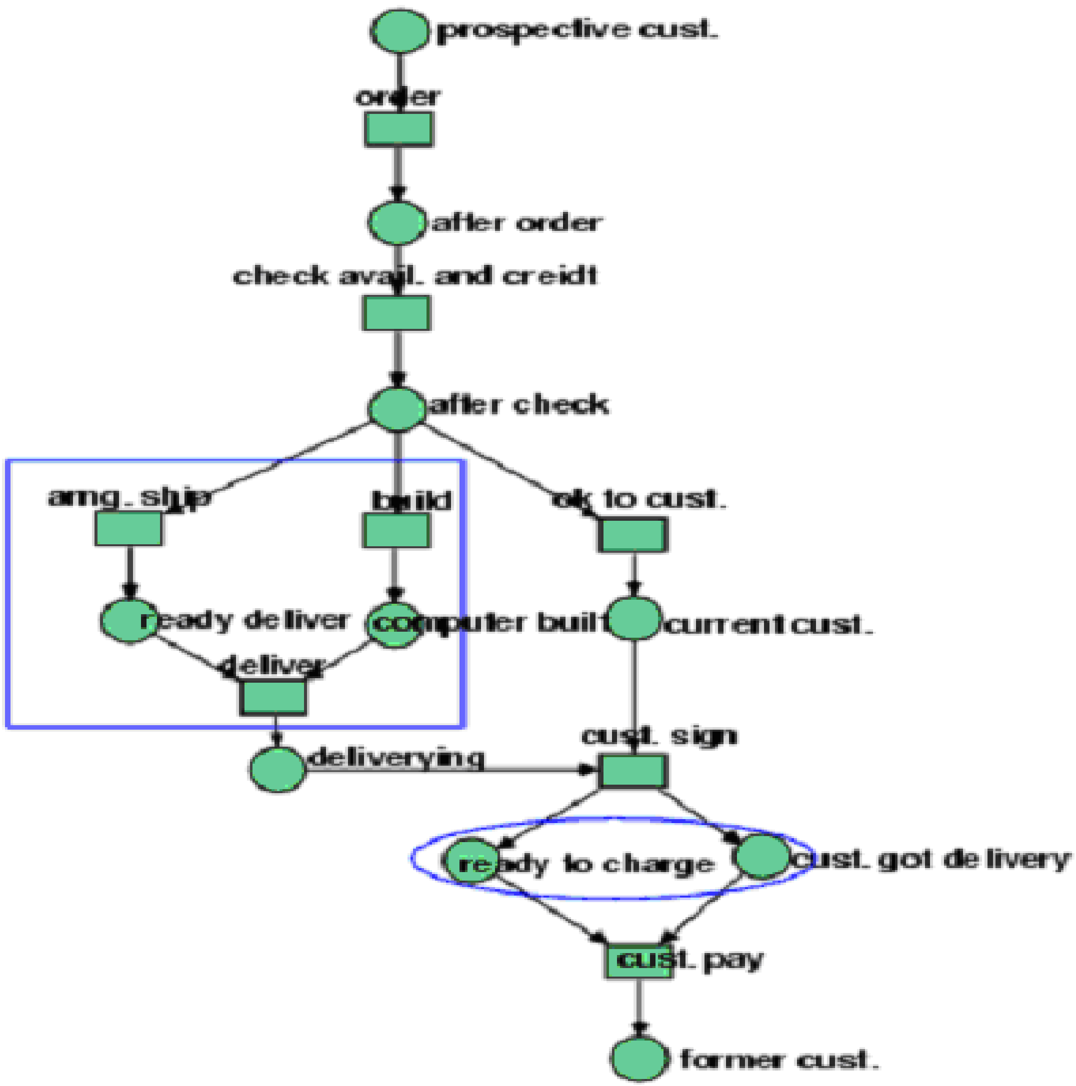
Online Computer Shopping System



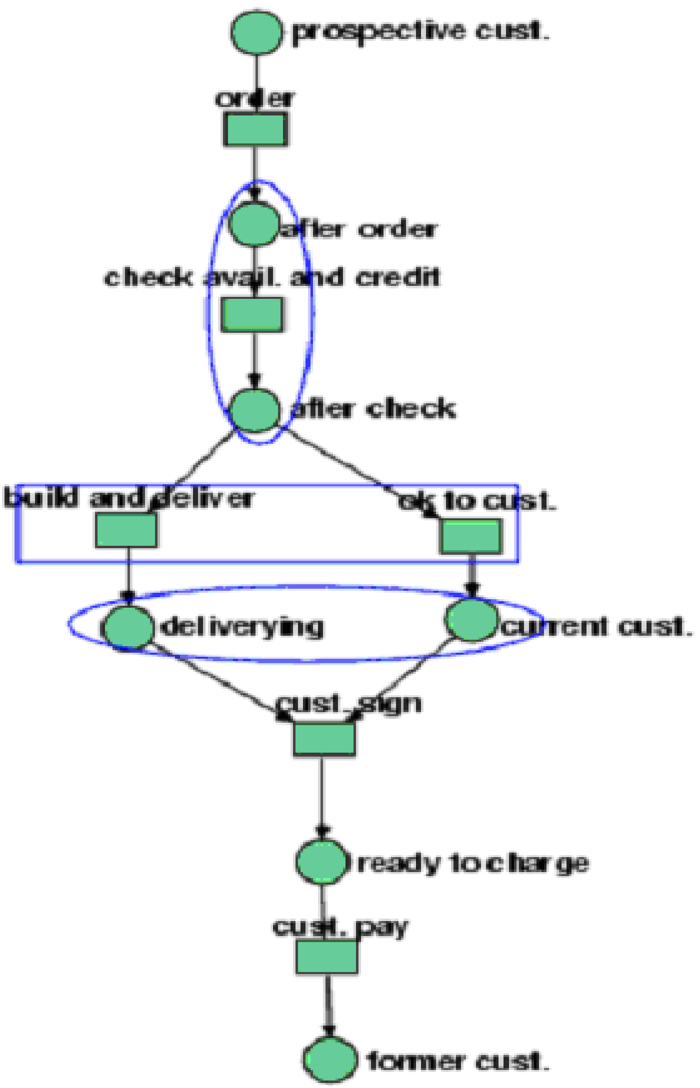
Petri Net Model of OCSS



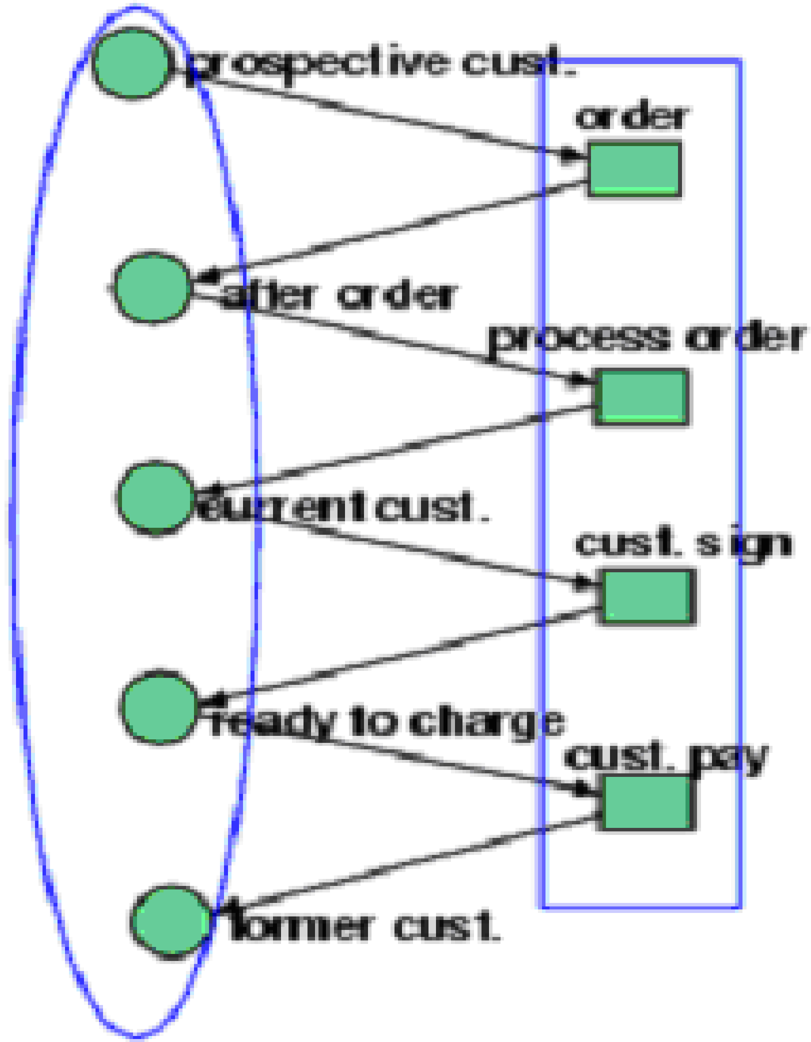
Abstraction from Customer Pers



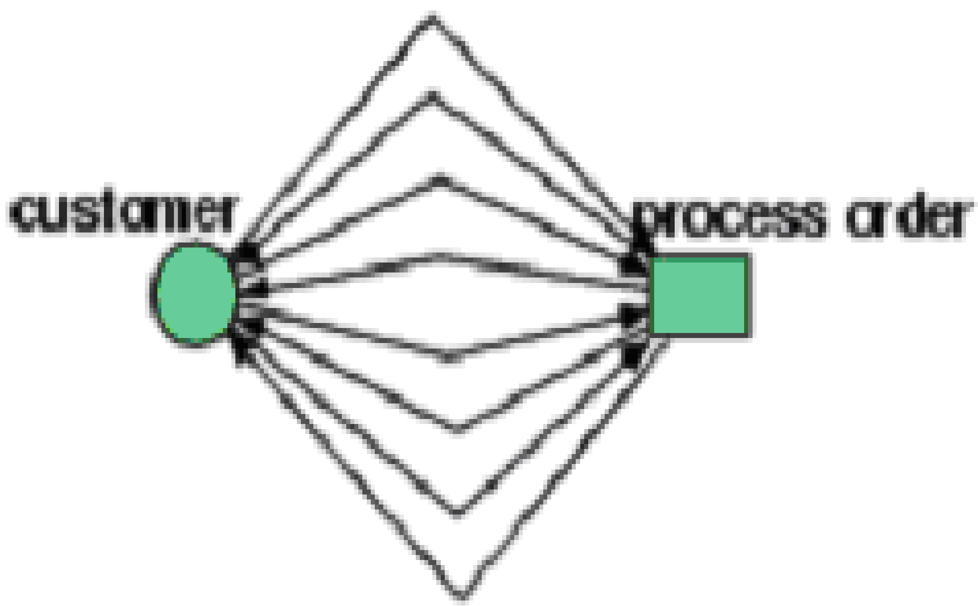
Abstraction from Customer Perspective



Abstraction from Customer Perspective



Highest Level of Abstraction of OCSS from Customer Perspective



Conclusions

- **Petri net morphisms** are well suited for conceptual modeling, specification, analysis and verification of concurrent systems by preserving desired structural and behavioral properties of the system at different levels of abstraction

Future work

- The **automation problem** of applying Petri net morphisms in the development of concurrent systems (project with Swapnil Chaudhuri)
- **Integrate Petri net morphisms** into Petri net based software development tools like Renew or PEP (project with Swapnil Chaudhuri)
- study of morphisms' properties for Colored Petri nets
- Study Petri net morphisms with respect to some **place and transition invariants** of Petri nets
- Apply Petri net morphisms to **other application domains**
- Use Petri net morphisms in **analysis and verification of Petri nets**

References

[AAH98] Adam, N.R.; Atluri, V.; Huang, W.K., Modeling and Analysis of Workflows using Petri nets, *Journal of Intelligent Information Systems*, *Vol. 10: No. 2*, pp.131-158, 1998.

[Ber87] Berthelot, G., Transformations and Decompositions of Nets, *LNCS 254: Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part I, Proc. of an Advanced Course, Bad Honnef, Sep.1986*, pp.359-376, Springer-Verlag, 1987.

[BB99] Bednarczyk, M. A.; Borzyszkowski, A. M. , General Morphisms of Petri nets, *LNCS, 1644: Automata, Languages and Programming*, pp.190-199, Springer-Verlag, 1999.

[CZ98] Cheung, T. Y.; Zeng, W., Invariant-preserving Transformations for the Verification of Place/Transition Systems, *IEEE Trans. on Systems, Man and Cybernetics,* pp.114-121, 1998.

[DKR97] Devillers, R.; Klaudel, H.; Riemann, R. C., General Refinement for High level Petri Nets, *LNCS 1346: Proc.* of *Foundations of Software Technology and Theoretical Computer Science '97,* pp.297-311, Springer-Verlag, 1997.

[DM96] Desel, J.; Merceron, A., Vicinity Respecting Homomorphisms for Abstracting System Requirements, *Bericht 337, Institut AIFB, Universit\"at Karlsruhe*, 1996.

[Feh91] Fehling, R., A Concept of Hierarchical Petri Nets with Building Blocks, *LNCS 674: Advances in Petri Nets 1993*, pp.148-168, Springer-Verlag, 1993.

References, ctnd.

[GV03] Girault, C., Valk, R., Petri Nets for Systems Engineering, A Guide to Modeling, Verification, and Applications, Springer, 2003.

[Lee87] Lee, H. K., Generalized Petri Net Reduction Method, *IEEE Transactions on Systems, Man and Cybernetics, SMC-17: No.2*, pp.297-302, 1987.

[MW03-1] **Mikolajczak, B., Wang, Z**., Conceptual Modeling of Concurrent Information Systems with General Morphisms of Petri nets, *LNCS, Intelligent Information Processing and Web Mining, Proc. of the International IIS: IIPWM′03 Conference, Zakopane, Poland, June 2-5, 2003*, pp.535-540, Physica Series, Springer-Verlag, 2003.

[MW03-2] **Mikolajczak, B., Wang Z**., Conceptual Modeling of Concurrent Systems through Stepwise Abstraction and Refinement using Petri Net Morphisms, Proc. of the Int. Conference on Conceptual Modeling, ER-2003, Oct. 13-16, 2003, Chicago, Illinois, LNCS 2813, pp. 433-445, Springer Verlag, 2003.

[MW04-1] **Mikolajczak, B.., Wang Z.,** Stepwise Abstractions of Workflow Systems using Petri net Morphisms, Proc. of the 2004 IEEE Int. Conference on Networking, Sensing and Control, IEEE ICNCS 2004, March 21-23, 2004, Taipei,Taiwan, accepted for publication.

References, ctnd.

[PGE98] Padberg, J.; Gajewsky, M.; Ermel, C. Rule-based Refinement of High-level Nets Preserving Safety Properties, *LNCS 1382: Fundamental Approaches to Software Engineering,* pp.221-238, Springer-Verlag, 1998.

[Rei98] Reisig, W. Elements of Distributed Algorithms: Modeling and Analysis with Petri Nets, Springer-Verlag, 1998.

[Van97] Van Der Aalst, W. M. P., Verification of Workflow Nets, *LNCS 1248*, pp.407-426, Springer-Verlag, 1997.

[Van98] Van Der Aalst, W. M. P., The Application of Petri Nets to Workflow Management, *Journal of Circuits, Systems and Computers, Vol.8: No. 1*, pp.21-66, 1998.

[Win87] Winskel, G., Petri nets, Algebras, Morphisms and Compositionality, *Information & Computation, Vol. 72: No. 3*, pp.197-238, 1987.

Thank you for your attention !!!