Petri Hypernets
for modeling mobile agents

Lucia Pomello

Università degli Studi di Milano - Bicocca
Dipartimento di Informatica, Sistemistica e Comunicazione

Moscow, 14/04/2014

Petri Hypernets for modeling mobile agents

Joint work with:

Marek Bednarczyk
Wiesław Pawłowski

Polish Academy of Science
IPI-PAN Gdańsk

Luca Bernardinello
Marco Mascheroni *

Universitá degli Studi
di Milano–Bicocca

Petri Hypernets for modeling mobile agents

Joint work with:

Marek Bednarczyk
Wiesław Pawłowski

Polish Academy of Science
IPI-PAN Gdańsk

Luca Bernardinello
Marco Mascheroni *

Universitá degli Studi
di Milano–Bicocca

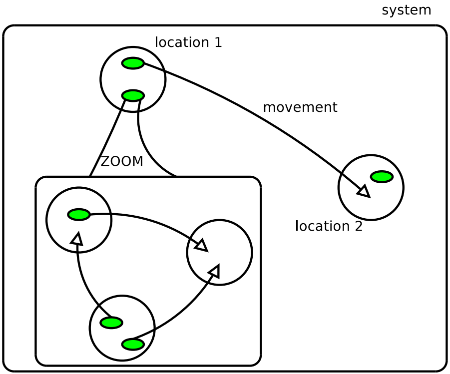
* now at CERN
(Centre Européen Recherche Nucléaire)

Petri Hypernets for modeling mobile agents

Outline

1. Systems of mobile agents
2. Petri Hypernets, the model
3. from Petri Hypernets to a basic class of Petri nets
4. Generalized hypernets
5. Two case studies
6. Conclusions

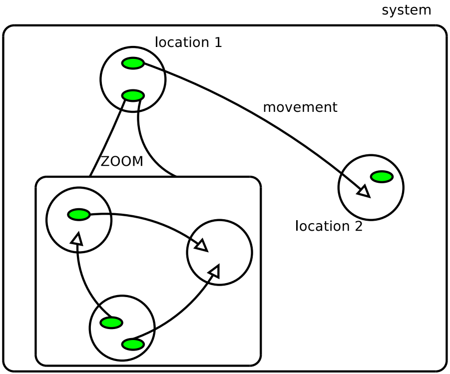
Systems of Mobile Agents



Main concepts

- Agents are *autonomous* and *interacting*
- They are embedded into *locations*
- *Mobility* means an agent can change its location
- *Hierarchical* structure which may *change*

Systems of Mobile Agents



Main concepts

- Agents are *autonomous* and *interacting*
- They are embedded into *locations*
- *Mobility* means an agent can change its location
- *Hierarchical* structure which may change

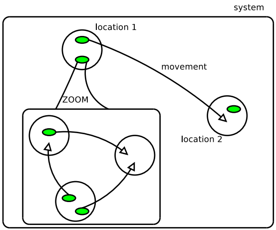
Main aim

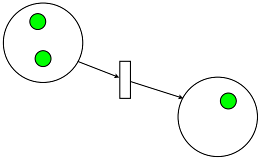
- A *formal* model based on *Petri net theory* for systems of mobile agents

The Nets-within-nets Paradigm[1] and Mobile Agents

The Nets-within-nets Paradigm

- Petri Nets $\Longrightarrow$ Nets-within-nets
- Tokens of a Petri net can be nets themselves





Modeling Mobile Agents

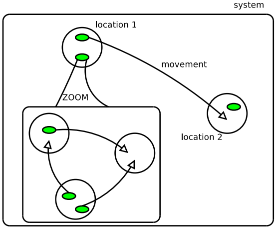
- Agents modeled as *nets*, which are *tokens* inside other agents
- Environment is a special agent
- Hierarchy of nets/agents

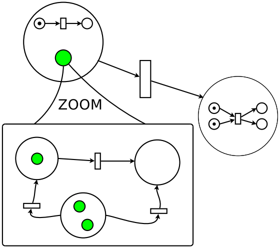
[1] R. Valk, Hamburg Univ., 1996

The Nets-within-nets Paradigm[1] and Mobile Agents

The Nets-within-nets Paradigm

- Petri Nets $\Longrightarrow$ Nets-within-nets
- Tokens of a Petri net can be nets themselves



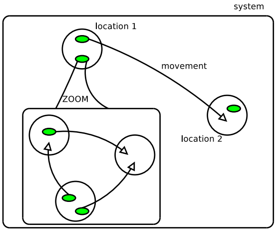


Modeling Mobile Agents

- Agents modeled as *nets*, which are *tokens* inside other agents
- Environment is a special agent
- Hierarchy of nets/agents

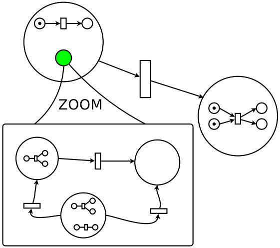
[1] R. Valk, Hamburg Univ., 1996

The Nets-within-nets Paradigm[1] and Mobile Agents



The Nets-within-nets Paradigm

- Petri Nets $\Longrightarrow$ Nets-within-nets
- Tokens of a Petri net can be nets themselves



Modeling Mobile Agents

- Agents modeled as *nets*, which are *tokens* inside other agents
- Environment is a special agent
- Hierarchy of nets/agents

[1] R. Valk, Hamburg Univ., 1996

SECTION 2

Petri Hypernets, the model

Petri Hypernets – introduction

$$< \mathcal{N}, \mu >$$

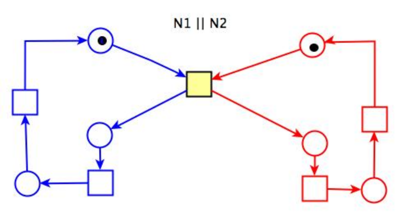
- Hypernet $\mathcal{N}$: a set of *agents*
- Agent A: a *net*, synchronous product of state-machine nets (*modules*)

Petri Hypernets – introduction

$$< \mathcal{N}, \mu >$$

- Hypernet $\mathcal{N}$: a set of *agents*
- Agent A: a *net*, synchronous product of state-machine nets (*modules*)

synchronization: merging transitions



Petri Hypernets – introduction

$$< \mathcal{N}, \mu >$$

- Hypernet $\mathcal{N}$: a set of *agents*
- Agent A: a *net*, synchronous product of state-machine nets (*modules*)
- Each agent is located *in a place of another agent*, and can move around (inside a state machine component)

Petri Hypernets – introduction

$$< \mathcal{N}, \mu >$$

- Hypernet $\mathcal{N}$: a set of *agents*
- Agent A: a *net*, synchronous product of state-machine nets (*modules*)
- Each agent is located *in a place of another agent*, and can move around (inside a state machine component)
- Typing

  σ_1 : Agents $\rightarrow$ Sorts

  σ_2 : Modules $\rightarrow$ Sorts $\qquad$ ($\Longrightarrow$ $\quad$ Places $\rightarrow$ Sorts)

Petri Hypernets – introduction

$$< \mathcal{N}, \mu >$$

- Hypernet $\mathcal{N}$: a set of *agents*
- Agent A: a *net*, synchronous product of state-machine nets (*modules*)
- Each agent is located *in a place of another agent*, and can move around (inside a state machine component)
- Typing
  σ_1 : Agents $\rightarrow$ Sorts
  σ_2 : Modules $\rightarrow$ Sorts ($\Longrightarrow$ Places $\rightarrow$ Sorts)
- Hypermarking $\mu : \mathcal{N} \rightarrow \mathcal{P}_{\mathcal{N}}$ (*compatible* with the typing)
- *Tree-like* containment hierarchy.

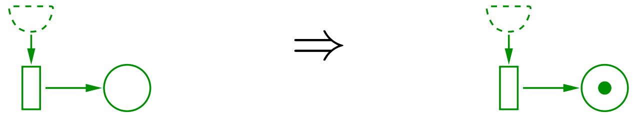
The agent hierarchy may change: *virtual places*

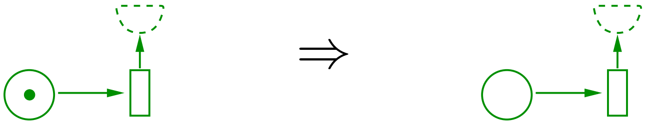
Virtual places specify how tokens may change level in the hierarchy

The agent hierarchy may change: *virtual places*

Virtual places specify how tokens may change level in the hierarchy

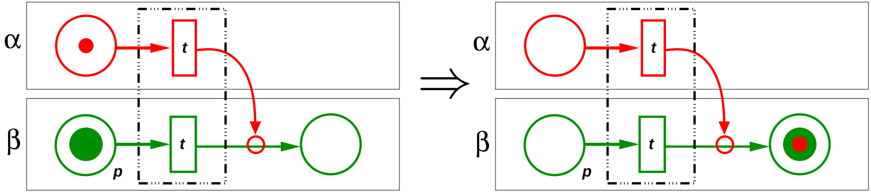
1. Between an agent and its super-agent





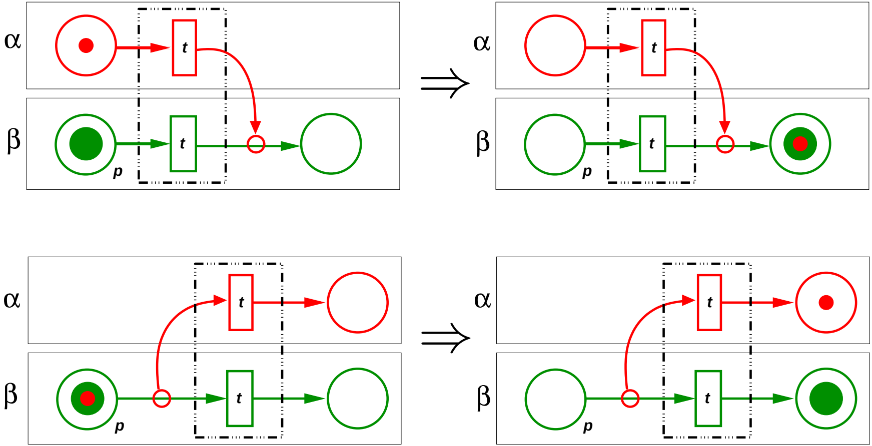
Virtual places

2. Within the same agent: inter-module exchange of tokens



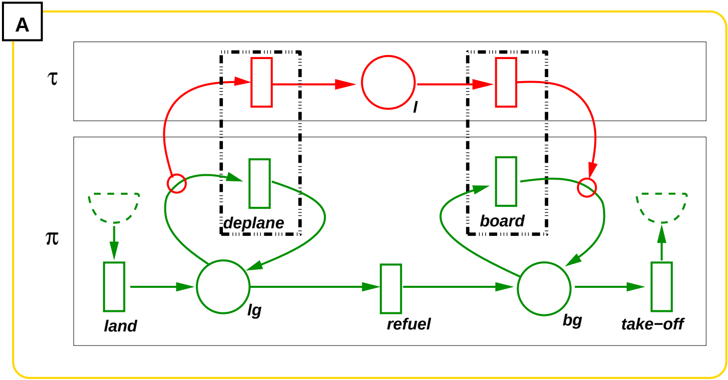
Virtual places

2. Within the same agent: inter-module exchange of tokens



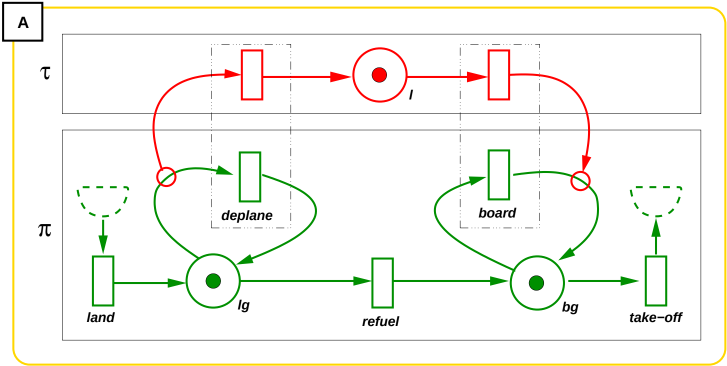
Example – modelling an airport

- An airport agent A with modules for planes (π) and travellers (τ)



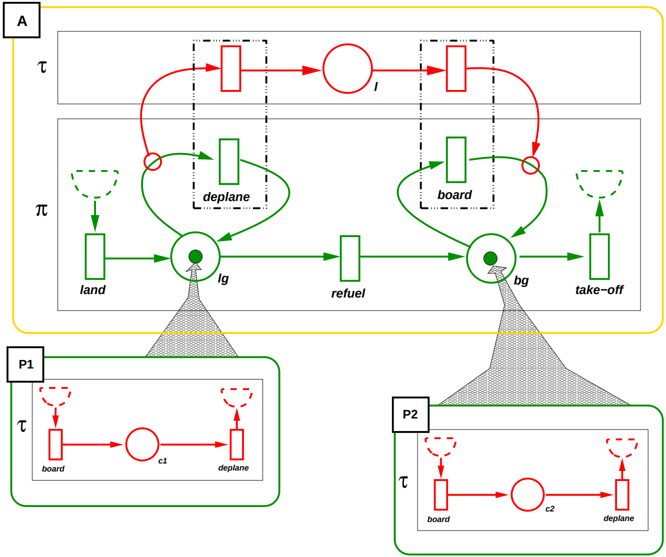
Example – modelling an airport

- An airport agent A with modules for planes (π) and travellers (τ)



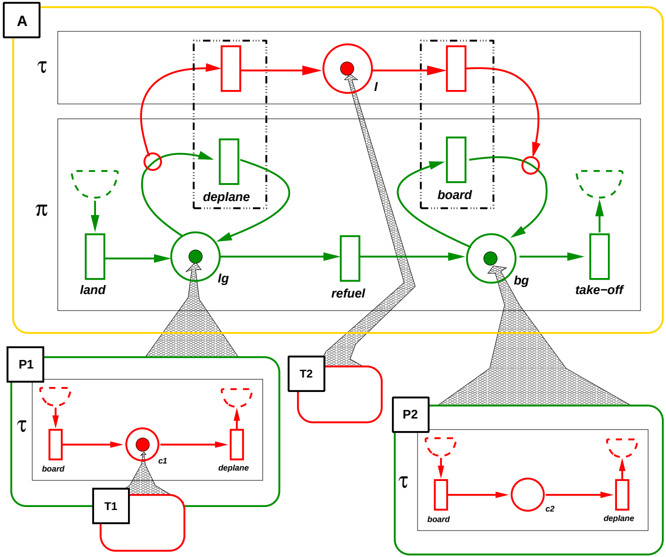
Example – modelling an airport

- An airport agent A with modules for planes (π) and travellers (τ)



Example – modelling an airport

- An airport agent A with modules for planes (π) and travellers (τ)



Dynamics—building a consortium

The state of the system (i.e., the hypermarking) changes by firing consortia

A *consortium* is made by:

- the name of a transition t
- a set of *active agents* (containing t) moving other agents
- a set of *passive agents* being moved by their *super-agents*

Remark: an agent can be both *active* and *passive*

Dynamics—building a consortium

The state of the system (i.e., the hypermarking) changes by firing consortia

A *consortium* is made by:

- the name of a transition t
- a set of *active agents* (containing t) moving other agents
- a set of *passive agents* being moved by their *super-agents*

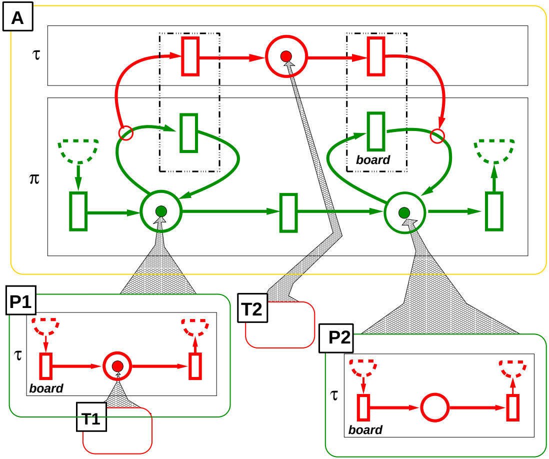
Remark: an agent can be both *active* and *passive*

Firing a consortium

the t transitions are fired *synchronously*

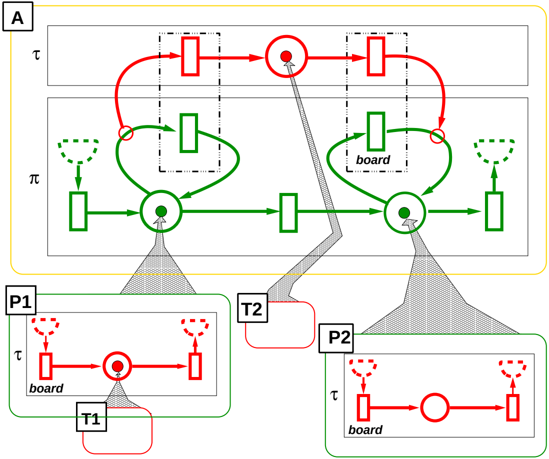
Dynamics—firing a consortium

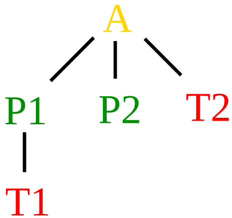
board is enabled



Dynamics—firing a consortium

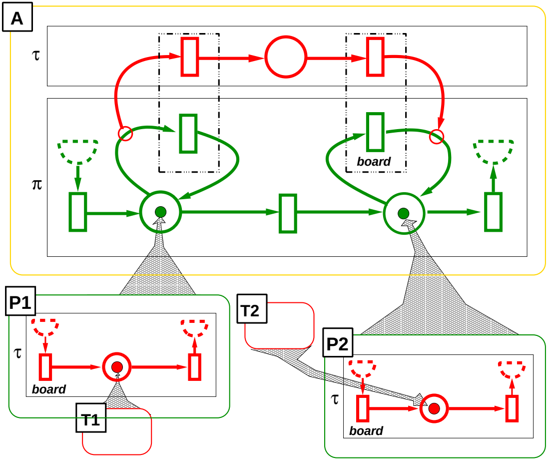
board is enabled





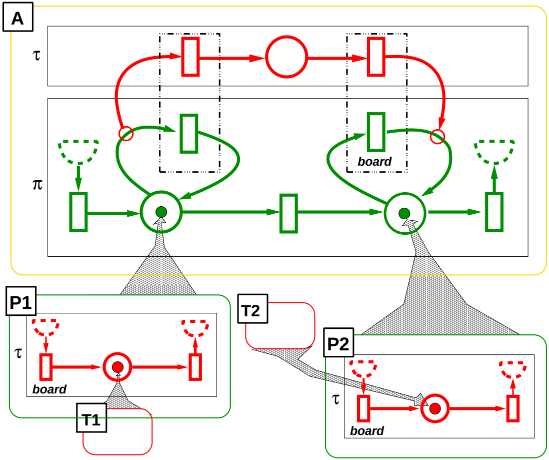
Dynamics—firing a consortium

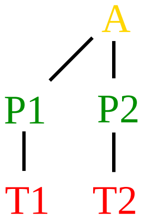
after the firing of *board*



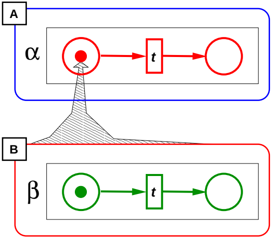
Dynamics—firing a consortium

after the firing of *board*

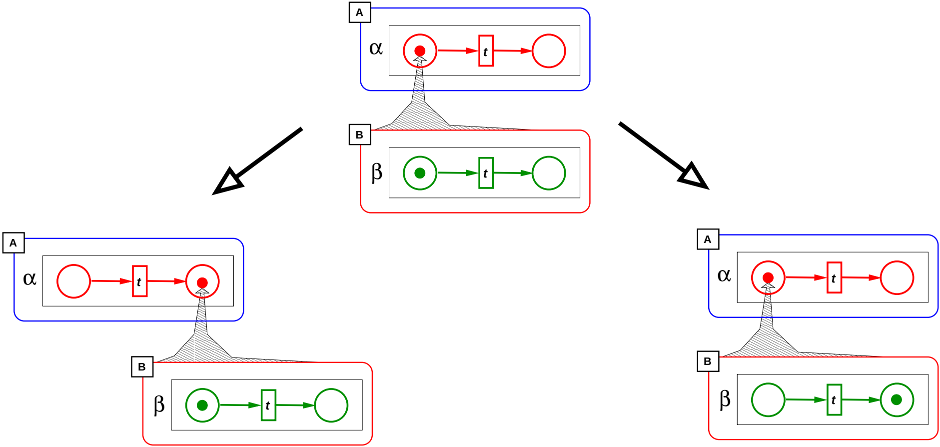




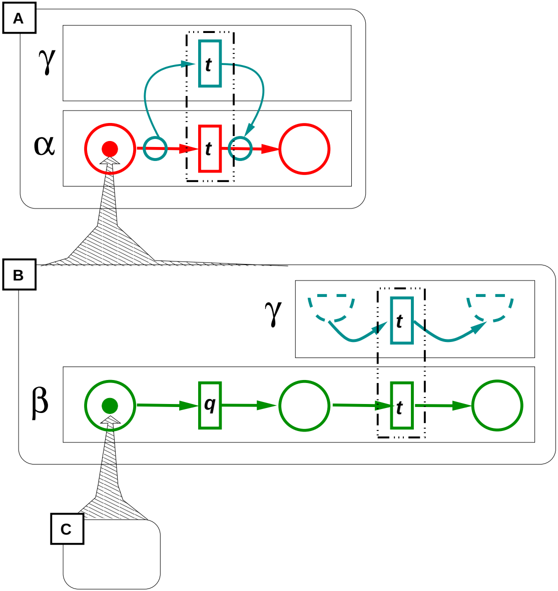
Two independent consortia



Two independent consortia



Inter-level synchronization via short loop



Remarks

boundedness

Petri Hypernets are such that there is *no* creation *no* destruction of agents

Remarks

boundedness

Petri Hypernets are such that there is *no* creation *no* destruction of agents

Theorem: Tree-like Structure Preservation

Let $< \mathcal{N}, \mu >$ be a Petri Hypernet, $\mathcal{M}$ a hypermarking, Γ a consortium enabled in $\mathcal{M}$, and $\mathcal{M}'$ be the hypermarking reached after executing Γ.

If the agent's hierarchy induced by $\mathcal{M}$ is a *tree*, then the agent's hyerarchy induced by $\mathcal{M}'$ is also a *tree*.

SECTION 3

From Petri Hypernets to a basic Petri net class (1-safe nets)

From Petri Hypernets to 1-safe nets

$$\mathcal{F} : < \mathcal{N}, \mu > \longrightarrow N = (B, E, F, M_\mu)$$

From Petri Hypernets to 1-safe nets

$$\mathcal{F} : < \mathcal{N}, \mu > \longrightarrow N = (B, E, F, M_\mu)$$

- B *conditions*: $< p, A >$ "agent A is in place p";

From Petri Hypernets to 1-safe nets

$$\mathcal{F} : < \mathcal{N}, \mu > \longrightarrow N = (B, E, F, M_\mu)$$

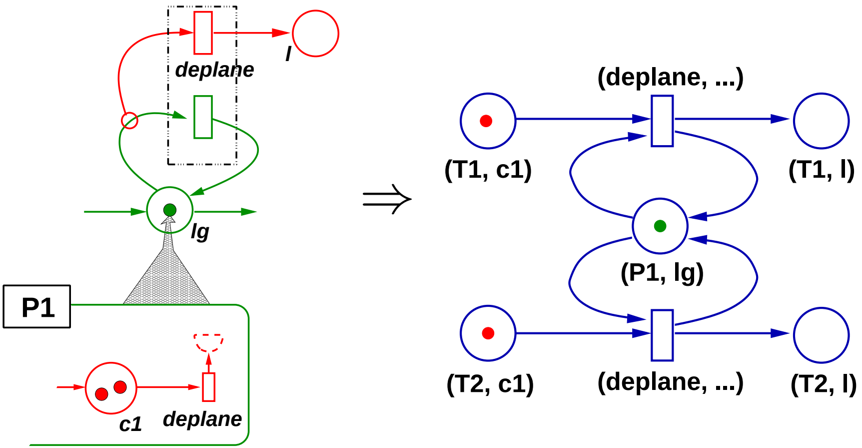
- B *conditions*: $< p, A >$ "agent A is in place p";
- E *events*: "potential consortia";

From Petri Hypernets to 1-safe nets

$$\mathcal{F} : < \mathcal{N}, \mu > \longrightarrow N = (B, E, F, M_\mu)$$

- B *conditions*: $< p, A >$ "agent A is in place p";
- E *events*: "potential consortia";
- $< p, A > \in M_\mu \iff \mu(A) = p$.

From Petri Hypernets to 1-safe nets – an example



From Petri Hypernets to 1-safe nets

$$\mathcal{F} : < \mathcal{N}, \mu > \longrightarrow N = (B, E, F, M_\mu)$$

N is such that:

- each event is *balanced* (i.e.: $|pre(e)| = |post(e)|$);
- N is *1-safe*;

From Petri Hypernets to 1-safe nets

$$\mathcal{F} : < \mathcal{N}, \mu > \longrightarrow N = (B, E, F, M_\mu)$$

N is such that:

- each event is *balanced* (i.e.: $|pre(e)| = |post(e)|$);
- N is *1-safe*;
- N is *state machine decomposable*;

From Petri Hypernets to 1-safe nets

$$\mathcal{F} : < \mathcal{N}, \mu > \longrightarrow N = (B, E, F, M_\mu)$$

N is such that:

- each event is *balanced* (i.e.: $|pre(e)| = |post(e)|$);
- N is *1-safe*;
- N is *state machine decomposable*;
- the *hyper-marking graph* of $< \mathcal{N}, \mu >$ is isomorphic to the *case graph* of N.

Hypermarkings of $< \mathcal{N}, \mu >$ and reachable markings of N

Theorem

A consortium Γ is enabled at a hypermarking $\mathcal{M}$ of $< \mathcal{N}, \mu >$ iff the corresponding transition t_Γ is enabled at the marking $f(\mathcal{M})$ of N.

The execution of t_Γ leads to a marking which corresponds to the hypermarking reached after executing Γ

Hypermarkings of $<\mathcal{N}, \mu>$ and reachable markings of N

Theorem

A consortium Γ is enabled at a hypermarking $\mathcal{M}$ of $<\mathcal{N}, \mu>$ iff the corresponding transition t_Γ is enabled at the marking $f(\mathcal{M})$ of N.

The execution of t_Γ leads to a marking which corresponds to the hypermarking reached after executing Γ

Consequences

- All the analysis techniques available for 1-safe nets can be applied to Petri Hypernets;
- all the properties decidable for 1-safe nets are decidable for Hypernets, too.

From Petri Hypernets to 1-safe nets

Analysis of a Hypernet through the analysis of the 1-safe net

- place-invariants:

  - some specific ones identify the structural components of the Hypernet
  (for ex.: all places where an agent can be hosted)

From Petri Hypernets to 1-safe nets

Analysis of a Hypernet through the analysis of the 1-safe net

- place-invariants:

  - some specific ones identify the structural components of the Hypernet
  (for ex.: all places where an agent can be hosted)

  - others are not so easily identifiable
  (for ex.: they contain places from different modules / agents)

- . . .

From Petri Hypernets to 1-safe nets

Analysis of a Hypernet through the analysis of the 1-safe net

- place-invariants:

  - some specific ones identify the structural components of the Hypernet
  (for ex.: all places where an agent can be hosted)

  - others are not so easily identifiable
  (for ex.: they contain places from different modules / agents)

- . . .

- "In the future":
  define invariants and other analysis technique directly on Hypernets.

Further work on Petri Hypernets

- Agent Aware Transition systems, as a behavioral model of mobile agents and as a basis for model checking Petri Hypernets. [2]
- A logic for specifying temporal and structural properties of Petri Hypernets, towards model checking (by the IPI-PAN group). [3]

[2] M. A. Bednarczyk, L. Bernardinello, W. Pawłowski, L. Pomello, Modelling and Analysing Systems of Agents by Agent-aware Transition Systems, 2008

[3] M. A. Bednarczyk, W. Jamroga, W. Pawlowski, Expressing and Verifying Temporal and Structural Properties of Mobile Agents. In *Fundam. Inform.*, 72(1-3): 51-63, 2006.

SECTION 4

From Petri Hypernets to Generalized hypertnets

From Petri Hypernets to Generalized hypertnets [4]

Motivations

Petri Hypernets have some constraints which can be too strict, e.g.:

- typing of agents,
- the modular structure of agents, ...

[4] Marco Mascheroni's PhD thesis, University of Milano -Bicocca, 2011

From Petri Hypernets to Generalized hypertnets [4]

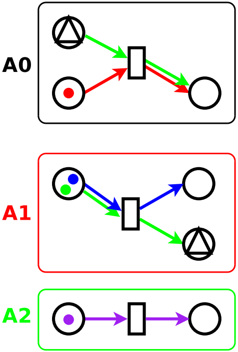
Motivations

Petri Hypernets have some constraints which can be too strict, e.g.:

- typing of agents,
- the modular structure of agents, ...

New Features

- Use of *paths* ⇒ weighted arcs
- Multi-typed agents
- A more flexible synchronization between adjacent agents



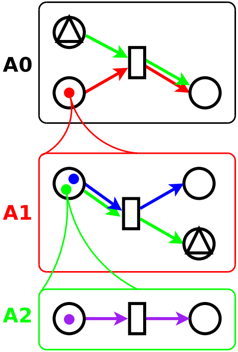
[4] Marco Mascheroni's PhD thesis, University of Milano -Bicocca, 2011

Generalized hypernet Consortia

The state of the system (i.e., the hypermarking) changes by firing consortia

Elements of a Consortium

- A set of transitions (with the same label is chosen)

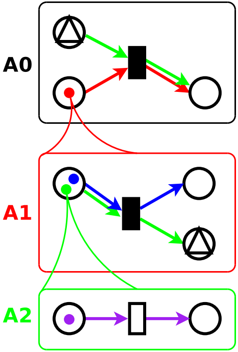


Generalized hypernet Consortia

The state of the system (i.e., the hypermarking) changes by firing consortia

Elements of a Consortium

- A set of transitions (with the same label is chosen)

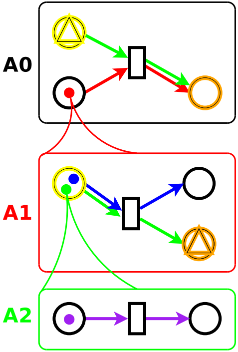


Generalized hypernet Consortia

The state of the system (i.e., the hypermarking) changes by firing consortia

Elements of a Consortium

- A set of transitions (with the same label is chosen)
- Binding virtual places

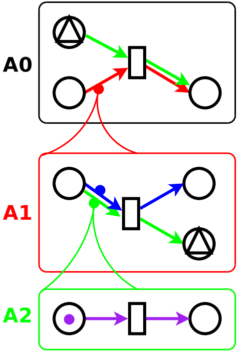


Generalized hypernet Consortia

The state of the system (i.e., the hypermarking) changes by firing consortia

Elements of a Consortium

- A set of transitions (with the same label is chosen)
- Binding virtual places
- Agent/Path association

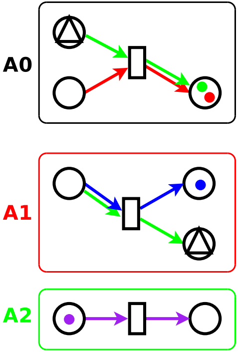


Generalized hypernet Consortia

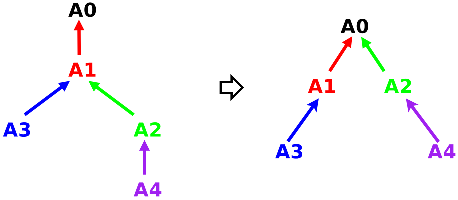
The state of the system (i.e., the hypermarking) changes by firing consortia

Elements of a Consortium

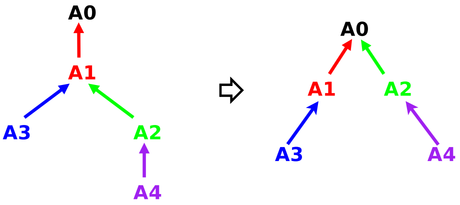
- A set of transitions (with the same label is chosen)
- Binding virtual places
- Agent/Path association
- The consortium can fire

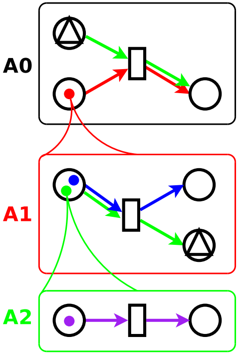


Hypermarking graph of Generalized hypernets

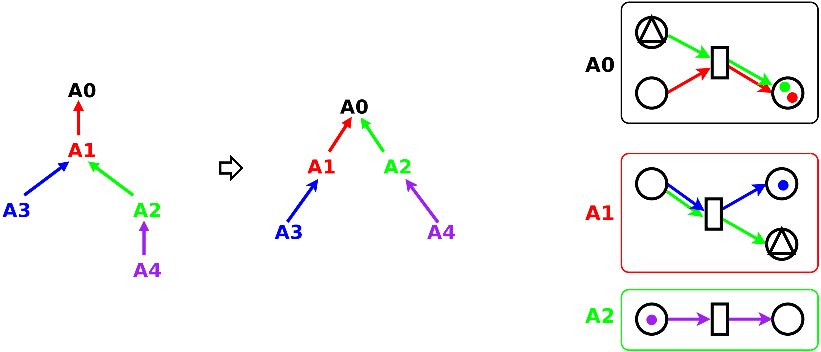


Hypermarking graph of Generalized hypernets





Hypermarking graph of Generalized hypernets



Generalized Hypernets: further results

- From Generalized Hypernets to 1-safe nets

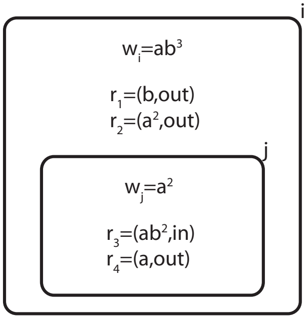
Generalized Hypernets: further results

- From Generalized Hypernets to 1-safe nets
- Definition of Unfolding of a Generalized Hypernet

SECTION 5

Two case studies

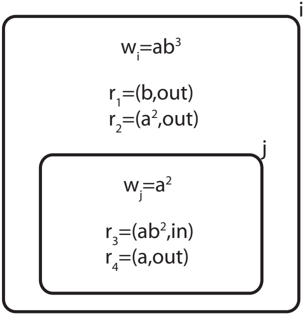
5.a) Modeling a Class of Membrane Systems



P Systems with Symport-Antiport Rules

- Bioinspired computational models
- Nested membranes which contain molecules and rules
- Rules change the configuration of the system

5.a) Modeling a Class of Membrane Systems

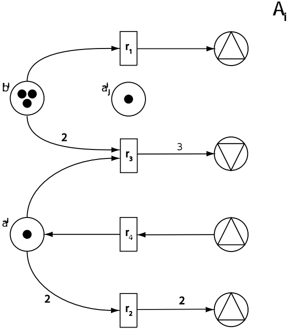


P Systems with Symport-Antiport Rules

- Bioinspired computational models
- Nested membranes which contain molecules and rules
- Rules change the configuration of the system

Modeling P systems

- Each membrane is an agent
- Molecules are tokens
- Rules are transitions
- Bijection *g* between hypermarkings and configurations



Modeling a Class of P Systems [5]

Theorem

Let Π be a P system, and let $< \mathcal{N}, \mu >$ be the Hypernet which models it.

If a set of rules ρ is enabled in a configuration C of Π, **then** the set of consortia U, which correspond to ρ, are enabled in the hypermarking $g(C)$.

If C' is the configuration reached after executing ρ, **then** $g(C')$ is the hypermarking reached after firing U.

[5] L. Bernardinello, N. Bonzanni, M. Mascheroni, L. Pomello. Modeling symport/antiport p systems with a class of hierarchical Petri nets. In *Membrane Computing*, of *LNCS 4860*, p. 124–137, 2007..

Modeling a Class of P Systems [5]

Theorem

Let Π be a P system, and let $< \mathcal{N}, \mu >$ be the Hypernet which models it.

If a set of rules ρ is enabled in a configuration C of Π, **then** the set of consortia U, which correspond to ρ, are enabled in the hypermarking $g(C)$.

If C' is the configuration reached after executing ρ, **then** $g(C')$ is the hypermarking reached after firing U.

Possible Applications

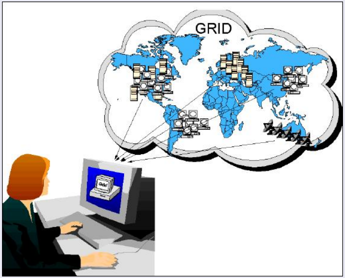
Then it is possible to use Petri nets analysis techniques to analyze P system.

[5] L. Bernardinello, N. Bonzanni, M. Mascheroni, L. Pomello. Modeling symport/antiport p systems with a class of hierarchical Petri nets. In *Membrane Computing*, of *LNCS 4860*, p. 124–137, 2007..

5.b) Modeling Grid Applications[6]

Grid Computing

- Analysis of data which are distributed in different data centers at different levels
- Final user should not care of data distribution
- Tools, which make easier the access to the Grid, are developed

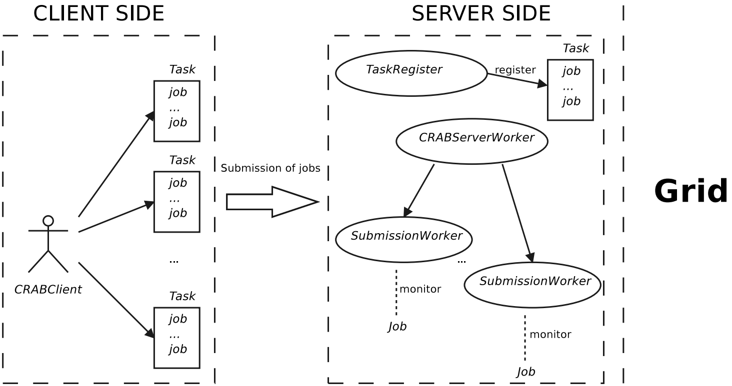


[6] M. Mascheroni, F. Farina, Nets within nets paradigm and grid computing. In ToPNoC, p. 201–220, 2012.

5.b) Modeling Grid Applications

- Is the Nets Within Nets Paradigm suitable to model/specify applications which use the Grid technology?
- Answer by modeling a Grid distributed data analysis tool serving the community of the Compact Muon Solenoid (CMS) experiment at the CERN Large Hadron Collider (LHC).

The CRAB Tool



- Split user's work into jobs and analyze datasets in parallel through the Grid
- CRAB (CMS Remote Analysis Builder) has a client-server architecture
- Sends/monitors/retrieves output of jobs from the Grid

Modeling Grid Applications

The submission use case of the CMS Remote Analysis Builder (CRAB) have been modeled by using Nets-Within-Nets, and in particular by using the tool RENEW (developed in Hamburg Univ.)

Results

Nets within nets seems to be a promising approach for modeling Grid applications:

- The hierarchical structure of the system is represented in the model;
- some bugs in the software have been identified;

Modeling Grid Applications

The submission use case of the CMS Remote Analysis Builder (CRAB) have been modeled by using Nets-Within-Nets, and in particular by using the tool RENEW (developed in Hamburg Univ.)

Results

Nets within nets seems to be a promising approach for modeling Grid applications:

- The hierarchical structure of the system is represented in the model;
- some bugs in the software have been identified;
- Marco Mascheroni got a job at CERN, ::)),
  he is now in the group developing a new version of the tool (CRAB3)

SECTION 6

Conclusions

Conclusions

Petri Hypernets: a summary

A hierarchical model for mobile agents in the *nets-within-nets* paradigm

- A Petri Hypernet is a collection of *modular* nets
  (obtained by composing *state machines* through transition synchronizations)

Conclusions

Petri Hypernets: a summary

A hierarchical model for mobile agents in the *nets-within-nets* paradigm

- A Petri Hypernet is a collection of *modular* nets
  (obtained by composing *state machines* through transition synchronizations)
- A Hypermarking defines a *hierarchy* on the nets
- The hierarchy dynamically changes

Conclusions

Petri Hypernets: a summary

A hierarchical model for mobile agents in the *nets-within-nets* paradigm

- A Petri Hypernet is a collection of *modular* nets
  (obtained by composing *state machines* through transition synchronizations)
- A Hypermarking defines a *hierarchy* on the nets
- The hierarchy dynamically changes
- Its semantics is given in terms of a basic net class

Conclusions

Petri Hypernets: a summary

A hierarchical model for mobile agents in the *nets-within-nets* paradigm

- A Petri Hypernet is a collection of *modular* nets
  (obtained by composing *state machines* through transition synchronizations)
- A Hypermarking defines a *hierarchy* on the nets
- The hierarchy dynamically changes
- Its semantics is given in terms of a basic net class
- They have been generalized and applied

Future work . . .

- relax some constraints of the model, e.g.:
  - add the possibility to create/destroy agents;
  - allow interactions among agents in the same location without the participation of their superagent

[7] [M. A. Bednarczyk, W. Jamroga, W. Pawlowski, Expressing and Verifying Temporal and Structural Properties of Mobile Agents. In *Fundam. Inform.*, 72(1-3): 51-63, 2006.

[8] M. Mascheroni, T. Wagner, L. Wustenberg.: Verifying reference nets by means of hypernets: a plugin for renew. In *Proc. PNSE'10*, p. 39–54, 2010..

Future work . . .

- relax some constraints of the model, e.g.:
  - add the possibility to create/destroy agents;
  - allow interactions among agents in the same location without the participation of their superagent
- develop techniques for property analysis, e.g.: Linear invariants, . . . ;

[7] [M. A. Bednarczyk, W. Jamroga, W. Pawlowski, Expressing and Verifying Temporal and Structural Properties of Mobile Agents. In *Fundam. Inform.*, 72(1-3): 51-63, 2006.

[8] M. Mascheroni, T. Wagner, L. Wustenberg.: Verifying reference nets by means of hypernets: a plugin for renew. In *Proc. PNSE'10*, p. 39–54, 2010..

Future work . . .

- relax some constraints of the model, e.g.:
  - add the possibility to create/destroy agents;
  - allow interactions among agents in the same location without the participation of their superagent
- develop techniques for property analysis, e.g.: Linear invariants, . . . ;
- define synchronous product of Petri (/ Genralized) hypernets;

[7] [M. A. Bednarczyk, W. Jamroga, W. Pawlowski, Expressing and Verifying Temporal and Structural Properties of Mobile Agents. In *Fundam. Inform.*, 72(1-3): 51-63, 2006.

[8] M. Mascheroni, T. Wagner, L. Wustenberg.: Verifying reference nets by means of hypernets: a plugin for renew. In *Proc. PNSE'10*, p. 39–54, 2010..

Future work . . .

- relax some constraints of the model, e.g.:
  - add the possibility to create/destroy agents;
  - allow interactions among agents in the same location without the participation of their superagent
- develop techniques for property analysis, e.g.: Linear invariants, . . . ;
- define synchronous product of Petri (/ Genralized) hypernets;
- develop a software tool, on the basis of already developed prototypes:
  - an *editor-simulator* of Hypernets with a model checker [7] (in Gdansk)
  - a plugin for Renew based on Hypernets (in collaboration with Hamburg Univ.). [8]

[7] [M. A. Bednarczyk, W. Jamroga, W. Pawlowski, Expressing and Verifying Temporal and Structural Properties of Mobile Agents. In *Fundam. Inform.*, 72(1-3): 51-63, 2006.

[8] M. Mascheroni, T. Wagner, L. Wustenberg.: Verifying reference nets by means of hypernets: a plugin for renew. In *Proc. PNSE'10*, p. 39–54, 2010..

Future work . . .

- relax some constraints of the model, e.g.:
  - add the possibility to create/destroy agents;
  - allow interactions among agents in the same location without the participation of their superagent
- develop techniques for property analysis, e.g.: Linear invariants, . . . ;
- define synchronous product of Petri (/ Genralized) hypernets;
- develop a software tool, on the basis of already developed prototypes:
  - an *editor-simulator* of Hypernets with a model checker [7] (in Gdansk)
  - a plugin for Renew based on Hypernets (in collaboration with Hamburg Univ.). [8]
- a possible collaboration with the PAIS Lab combining *Nested Petri Nets* and *Peri Hypernets* ?

---

[7] [M. A. Bednarczyk, W. Jamroga, W. Pawlowski, Expressing and Verifying Temporal and Structural Properties of Mobile Agents. In *Fundam. Inform.*, 72(1-3): 51-63, 2006.

[8] M. Mascheroni, T. Wagner, L. Wustenberg.: Verifying reference nets by means of hypernets: a plugin for renew. In *Proc. PNSE'10*, p. 39–54, 2010..

References

- Marek A. Bednarczyk, Luca Bernardinello, Wiesław Pawłowski, and Lucia Pomello, Modelling Mobility with Petri Hypernets, In *WADT 2004*, pages 28 – 44, 2005.
- M. A. Bednarczyk, L. Bernardinello, W. Pawłowski, and L. Pomello, Modelling and Analysing Systems of Agents by Agent-aware Transition Systems, In: F. Fogelman-Souli, et alt., Mining Massive Data Sets for Security. p. 103-112, IOS Press, 2008.
- Marek A. Bednarczyk, Luca Bernardinello, Wiesław Pawłowski, and Lucia Pomello, From Petri hypernets to 1-safe nets, In *Proc. of the 4th Int. Workshop on Modelling of Objects, Components, and Agents, MOCA'06*, 2006.
- Marek A. Bednarczyk, Wojciech Jamroga, Wieslaw Pawlowski, Expressing and Verifying Temporal and Structural Properties of Mobile Agents. In *Fundam. Inform.*, 72(1-3): 51-63, 2006.
- M.A. Bednarczyk, W.Pawlowski, L.Bernardinello, L.Pomello, T.Borzyszkowski, A Multi-facet Approach to Dynamic Agent Systems. In *Fundam.Inform.*, 85(1-4):19-33,2008.
- Luca Bernardinello, Nicola Bonzanni, Marco Mascheroni, and Lucia Pomello. Modeling symport/antiport p systems with a class of hierarchical Petri nets. In *Membrane Computing*, of *LNCS 4860*, p. 124–137, 2007.
- Marco Mascheroni. Generalized hypernets and their semantics. In *Proc. of the Fith Int. Workshop on Modelling of Objects, Components and Agents, MOCA'09*, p.87–106, 2009.
- Marco Mascheroni, Thomas Wagner, Lars Wustenberg.Verifying reference nets by means of hypernets: a plugin for renew. In *Proc. of the Int. Workshop on Petri Nets and Software Engeneering PNSE'10*, p. 39–54, 2010.
- Marco Mascheroni, Hypernets: a Class of Hierarchical Petri Nets. PhD Thesis, Univerità degli Studi di Milano-Bicocca, 2011.
- Marco Mascheroni, Fabio Farina. Nets within nets paradigm and grid computing. In *Transaction on Petri Nets and Other Models of Concorrency*, p. 201–220, 2012.

THANK YOU
for your attention !

Спасибо большое !

GRAZIE!