Modular system development by composing Petri nets on interfaces

Lucia Pomello

Dipartimento di Informatica, Sistemistica e Comunicazione
University of Milano – Bicocca

Moscow, 21/04/2014
Joint work with

- Luca Bernardinello – DISCo, University of Milano–Bicocca
- Carlo Ferigato – JRC European Commission, Ispra
- Görkem Kılınc – Phd student, Univ. Milano – Bicocca
- Elisabetta Mangioni – Phd, now CNR, Milano
- Elena Monticelli – former Master Thesis student.
- Stefano Scacabarozzi – former Master Thesis student.
Modular system development by composing Petri nets on interfaces

Outline

1. Composing on interfaces, the intuition
2. \(\hat{N}\)-morphisms: abstraction and refinement
3. Composing on interfaces by \(\hat{N}\)-morphisms
4. Properties preserved/reflected
5. A new notion of morphisms: \(\alpha\)-morphisms
6. Application to modular synthesis
7. A case study
8. Conclusions
Section 1

Composing on interfaces, the intuition
Composition operations for Petri nets

\[ N = (P, T, F, M_0) \]

typical ways of composing nets

- synchronous
- asynchronous
- mixed
Synchronous Composition

merging transitions (synchronization)
Asynchronous Composition

merging places (channels)
Composing on interfaces, the intuition...

Elementary net systems \( N = (B, E, F, c_{in}) \):
- \( B \): conditions (boolean propositions) - \( E \): events

\[
\begin{align*}
N1 & \quad N2 \\
\text{p} & \quad \text{q}
\end{align*}
\]
Composing on interfaces, the intuition...

Elementary net systems $N = (B, E, F, c_{in})$:
- $B$: conditions (boolean propositions) - $E$: events
Composing on interfaces, the intuition...
Composing on interfaces, by means of morphisms

Lucia Pomello (DISCo)
Composing on interfaces, by means of morphisms

abstracting from details
Composing on interfaces, by means of **morphisms**

abstracting from details

\[
\text{if } q \text{ then } (s \land (r \lor u)) \lor w
\]
Composing on interfaces, by means of morphisms

N = N1 \[ N1 \] N2
Section 2

\(\hat{N}\)-morphisms: abstraction and refinement
the origins

**$N$-morphisms**

[ Nielsen, Rozenberg, Thiagarajan '92]

(intuitively) they *preserve* behaviours, i.e.:

if $\phi : N_1 \rightarrow N_2$ is a *$N$-morphism* then $N_2$ is *partially simulating* $N_1$. 
\( \hat{N} \)-morphisms for EN systems

\[ N_i = (B_i, E_i, F_i, c_{i,in}), \quad i = 1, 2 \]

\( \hat{N} \)-morphisms for EN systems \(^1\) \(^2\)

\[ N_i = (B_i, E_i, F_i, c_{i, in}), \; i = 1, 2 \]

\((\beta, \eta) : N_1 \rightarrow N_2\) is an \( \hat{N} \)-morphism iff

\( \beta \subseteq B_1 \times B_2 \) relation, \hspace{1cm} \( \eta : E_1 \rightarrow E_2 \) partial surjective map:

---


\(\hat{N}\)-morphisms for EN systems

\(N_i = (B_i, E_i, F_i, c_i, in), \ i = 1, 2\)

\((\beta, \eta) : N_1 \rightarrow N_2\) is an \(\hat{N}\)-morphism iff

- \(\beta \subseteq B_1 \times B_2\) relation,
- \(\eta : E_1 \rightarrow E_2\) partial surjective map:
  - \(\beta^{-1} : B_2 \rightarrow B_1\) total injective map
  - \(\forall (b_1, b_2) \in \beta : b_1 \in c_{1, in} \Leftrightarrow b_2 \in c_{2, in}\)

---

N-morphisms for EN systems

\[ N_i = (B_i, E_i, F_i, c_{i,in}), \quad i = 1, 2 \]

\[(\beta, \eta) : N_1 \rightarrow N_2 \text{ is an } \hat{N}\text{-morphism iff} \]

- \( \beta \subseteq B_1 \times B_2 \) relation,
- \( \eta : E_1 \rightarrow E_2 \) partial surjective map:

  - \( \beta^{-1} : B_2 \rightarrow B_1 \) total injective map
  - \( \forall (b_1, b_2) \in \beta : b_1 \in c_{1,in} \iff b_2 \in c_{2,in} \)
  - \( \eta(e_1) = e_2 \Rightarrow \beta(e_1) = e_2 \) and \( \beta(e_1^\bullet) = e_2^\bullet \)

\[ \]

\( \hat{N} \)-morphisms for EN systems  

\[ N_i = (B_i, E_i, F_i, c_{i,in}), \quad i = 1, 2 \]

\[ (\beta, \eta) : N_1 \rightarrow N_2 \] is an \( \hat{N} \)-morphism iff

- \( \beta \subseteq B_1 \times B_2 \) relation,
- \( \eta : E_1 \rightarrow E_2 \) partial surjective map:
  - \( \beta^{-1} : B_2 \rightarrow B_1 \) total injective map
  - \( \forall (b_1, b_2) \in \beta : b_1 \in c_{1,in} \iff b_2 \in c_{2,in} \)
  - \( \eta(e_1) = e_2 \Rightarrow \beta(\bullet e_1) = \bullet e_2 \) and \( \beta(e_1^\bullet) = e_2^\bullet \)
  - \( \eta(e_1) = \perp \Rightarrow \beta(\bullet e_1) = \beta(e_1^\bullet) = \emptyset \)

---

The counterimage of $N_I$, after $T$-simplification, is isomorphic to $N_I$.

$\hat{N}$-morphisms: refinement / abstraction
Section 3

Composing on interfaces by $\hat{N}$-morphisms
Composing two nets on an interface by $\hat{N}$-morphisms

\[
\begin{array}{c}
N_1 \\
\uparrow_{\beta_1, \eta_1} \\
N_I \\
\downarrow_{\beta_2, \eta_2} \\
N_2
\end{array}
\]
Composing two nets on an interface by $\hat{N}$-morphisms

$\hat{N}$-morphisms dictate the identification (composition) of elements
Composing two nets on an interface by $\hat{N}$-morphisms

$\hat{N}$-morphisms dictate the identification (composition) of elements

Proposition

The diagram commute
The composition: how to compose events

$N_1$ and $N_2$ are composed on the interface $i_1$. The composition is depicted as follows:

- $N_1$ with events $e_0$, $e_1$, and $e_2$ connected to $i_1$.
- $N_2$ with events $e_{00}$, $e_{01}$, $e_1$, $e_{20}$, and $e_{21}$ connected to $i_1$.
The composition: how to compose events

\[ N_1 < N_1 \rightarrow N_2 \]

\[ \langle e_1, e_1 \rangle < \langle e_2, e_0 \rangle < \langle e_2, e_0 \rangle < \langle e_2, e_2 \rangle < \langle e_2, e_2 \rangle < \langle e_2, e_2 \rangle \]
Composing on interfaces, by $\hat{N}$-morphisms

\[ N = N1[N1]N2 \]
Composition by $\hat{N}$-morphisms is **not** a Pullback

Example:

\[ N = N_1 [N_1] N_2 \]

---

Composition by \(\hat{N}\)-morphisms is \textbf{not} a Pullback

Example:

A pullback composition has been defined on a bit different morphisms/composition \(^3\)

Section 4

Properties preserved/reflected
Preserving/reflecting properties

refined system \( \varphi \rightarrow \) abstract system
Preserving/reflecting properties

- Refined system
- Abstract system

Diagram:

- Directed edge from "refined system" to "abstract system" labeled with \( \varphi \)
- Directed edge from "abstract system" to "refined system" labeled with "preserving"
Preserving/reflecting properties

\[ \text{refined system} \xrightarrow{\varphi} \text{abstract system} \]

preserving

reflecting
\( (\beta, \eta) : N_1 \to N_2 \)

- **S-invariants are reflected:**
  - if \( l_2 \) is an S-invariant of \( N_2 \), then \( l_1 = \beta^{-1}(l_2) \) is an S-invariant of \( N_1 \)
$\hat{N}$-morphisms: properties preserved/reflected

$$(\beta, \eta) : N_1 \rightarrow N_2$$

- S-invariants are reflected:
  if $I_2$ is an S-invariant of $N_2$, then $I_1 = \beta^{-1}(I_2)$ is an S-invariant of $N_1$

- S-invariants are **not** preserved
\hat{N}\text{-morphisms: properties preserved/reflected}

\((\beta, \eta) : N_1 \to N_2\)

- S-invariants are \textit{reflected}: if \(I_2\) is an S-invariant of \(N_2\), then \(I_1 = \beta^{-1}(I_2)\) is an S-invariant of \(N_1\)

- S-invariants are \textbf{not} preserved

- T-invariants are \textit{preserved}: if \(J_1\) is a T-invariant of \(N_1\), then \(J_2 = \eta(J_1)\) is a T-invariant of \(N_2\)
$\hat{N}$-morphisms: properties preserved/reflected

$(\beta, \eta) : N_1 \to N_2$

- S-invariants are **reflected**: if $I_2$ is an S-invariant of $N_2$, then $I_1 = \beta^{-1}(I_2)$ is an S-invariant of $N_1$

- S-invariants are **not** preserved

- T-invariants are **preserved**: if $J_1$ is a T-invariant of $N_1$, then $J_2 = \eta(J_1)$ is a T-invariant of $N_2$

- T-invariants are **not** reflected
Preserving properties

- It is possible that $N_I$, $N_1$, $N_2$ are live, but $N_1[N_I]N_2$ is not live;
- however, ...
reflecting sequences

Definition

\[ \text{FS}(N) \quad \text{firing sequences of } N, \]

\[ (\beta, \eta) : N \rightarrow N' \quad \hat{N}\text{-morphism} \]

\[ N \text{ reflects the sequences of } N' \text{ under } (\beta, \eta) \quad \text{iff} \]

\[ \forall v \in \text{FS}(N'), \exists w \in \text{FS}(N) \text{ such that: } \hat{\eta}(w) = v \]
If $N_1$ and $N_2$ reflect the sequences of $N_I$, respectively,
then $N_1[N_I]N_2$ reflects the sequences of $N_1$, $N_2$ and $N_I$. 
Deadlock-freeness?

\( N_I, N_A \) and \( N_B \) are deadlock-free and even live.
Deadlock-freeness?

\[ N_A[N_I]N_B \]
Deadlock-freeness?

\[ N_A[N_I]N_B \]

\[ N_A[N_I]N_B \text{ is dead} \]
Weak Bisimulation

if $N'$ and $N''$ are weakly bisimilar ($N' \approx^{BIS} N''$)

then $N'$ is deadlock-free iff $N''$ is deadlock-free

---

4 R. Milner, A Calculus of Communicating Systems, 1980
Weak Bisimulation\textsuperscript{4}

\begin{align*}
\text{if } N' \text{ and } N'' \text{ are weakly bisimilar } (N' \approx^{BIS} N'') \\
\text{then } N' \text{ is deadlock-free iff } N'' \text{ is deadlock-free}
\end{align*}

Remark
Weak Bisimulation is verified considering the reachability graphs.

\textsuperscript{4}R. Milner, A Calculus of Communicating Systems, 1980
Deadlock-freeness?

Theorem

\[ N_1 \approx^{BIS} N_1 \Rightarrow N \approx^{BIS} N_2 \]

Corollary

\[ (N_1 \approx^{BIS} N_I \land N_2 \text{ deadlock-free}) \Rightarrow N \text{ deadlock-free} \]
Deadlock-freeness?

Theorem

\[ N_1 \approx_{BIS} N_I \Rightarrow N \approx_{BIS} N_2 \]

if \( N' \approx_{BIS} N'' \) and \( N' \) is deadlock-free, then \( N'' \) is deadlock-free
Deadlock-freeness?

**Theorem**

\[ N_1 \approx^{BIS} N_I \Rightarrow N \approx^{BIS} N_2 \]

if \( N' \approx^{BIS} N'' \) and \( N' \) is deadlock-free, then \( N'' \) is deadlock-free

**Corollary**

\[ (N_1 \approx^{BIS} N_I \land N_2 \text{ deadlock-free}) \Rightarrow N \text{ deadlock-free} \]
Section 5

Refinement and composition based on a new notion of morphisms: $\alpha$-morphisms
Section 5

Refinement and composition based on a *new notion of morphisms*: $\alpha$-*morphisms*

for Elementary Net systems, covered by sequential components
Main aim

Refinement/abstraction and composition preserving/reflecting properties by considering behaviours, only locally
Main aim

Refinement/abstraction and composition preserving/reflecting properties by considering behaviours, only locally
Composition on Interfaces using $\alpha$-morphisms: the idea

$N_I$

---

5 Luca Bernardinello, Elisabetta Mangioni and Lucia Pomello, Local State Refinement and Composition of Elementary Net Systems: An Approach Based on Morphisms, ToPNoC VIII, 2013

Composition on Interfaces using $\alpha$-morphisms: the idea

$N_1 \xrightarrow{\varphi_1} N_I \xleftarrow{\varphi_2} N_2$

---

5 Luca Bernardinello, Elisabetta Mangioni and Lucia Pomello, Local State Refinement and Composition of Elementary Net Systems: An Approach Based on Morphisms, ToPNoC VIII, 2013

Composition on Interfaces using $\alpha$-morphisms: the idea

Luca Bernardinello, Elisabetta Mangioni and Lucia Pomello, Local State Refinement and Composition of Elementary Net Systems: An Approach Based on Morphisms, ToPNoC VIII, 2013

Properties preserved and reflected by $\alpha$-morphisms

$\varphi : N_1 \rightarrow N_2$ $\alpha$-morphism:

- “Good behaviour” of a ”bubble”
  - if an entering event to a bubble can fire then the bubble is empty
  - if an outgoing event from a bubble fires, it empties the bubble
Properties preserved and reflected by $\alpha$-morphisms

$\varphi : N_1 \rightarrow N_2$ $\alpha$-morphism :

- "Good behaviour" of a "bubble"
  - if an entering event to a bubble can fire then the bubble is empty
  - if an outgoing event from a bubble fires, it empties the bubble

- Sequential components are reflected
  - counter image of a sequential component is covered by sequential components
Properties preserved and reflected by $\alpha$-morphisms

$\varphi : N_1 \rightarrow N_2$ $\alpha$-morphism:

- “Good behaviour” of a ”bubble”
  - if an entering event to a bubble can fire then the bubble is empty
  - if an outgoing event from a bubble fires, it empties the bubble

- Sequential components are reflected
  - counter image of a sequential component is covered by sequential components

- Sequential components are not preserved
Properties preserved and reflected by $\alpha$-morphisms

$\varphi : N_1 \rightarrow N_2$ $\alpha$-morphism:

- “Good behaviour” of a “bubble”
  - if an entering event to a bubble can fire then the bubble is empty
  - if an outgoing event from a bubble fires, it empties the bubble

- Sequential components are reflected
  - counter image of a sequential component is covered by sequential components

- Sequential components are not preserved

- Reachable markings are preserved
Properties preserved and reflected by $\alpha$-morphisms

$\varphi : N_1 \rightarrow N_2$ $\alpha$-morphism:

- “Good behaviour” of a ”bubble”
  - if an entering event to a bubble can fire then the bubble is empty
  - if an outgoing event from a bubble fires, it empties the bubble

- Sequential components are reflected
  - counter image of a sequential component is covered by sequential components

- Sequential components are not preserved

- Reachable markings are preserved

- Reachable markings are reflected \textbf{iff}
  1. the initial marking of each bubble is at the start of the bubble itself
  2. no deadlock internal to a bubble and each final marking of a bubble enables the “same” set of events enabled by its image
Properties preserved and reflected by $\alpha$-morphisms

$\varphi : N_1 \rightarrow N_2$ $\alpha$-morphism:

- "Good behaviour" of a "bubble"
  - if an entering event to a bubble can fire then the bubble is empty
  - if an outgoing event from a bubble fires, it empties the bubble

- Sequential components are reflected
  - counter image of a sequential component is covered by sequential components

- Sequential components are not preserved

- Reachable markings are preserved

- Reachable markings are reflected iff
  1. the initial marking of each bubble is at the start of the bubble itself
  2. no deadlock internal to a bubble and each final marking of a bubble enables the "same" set of events enabled by its image
  3. a "Local unfolding" condition
Composition based on $\alpha$-morphisms, formal results

Proposition

$\alpha$-morphism: $N_1 \rightarrow N_I$ and $1 + 2 + 3$

$\Rightarrow$ $N_1$ is weakly bisimilar to $N_I$;
Composition based on $\alpha$-morphisms, formal results

Proposition

- $\alpha$-morphism: $N_1 \rightarrow N_I$ and $1 + 2 + 3$
  $\Rightarrow$ $N_1$ is weakly bisimilar to $N_I$;

- $\alpha$-morphism: $N_1 \rightarrow N_I$ and $1 + 2 + 3$ and $\alpha$-morphism: $N_2 \rightarrow N_I$
  $\Rightarrow$ $N_1\langle N_I \rangle N_2$ is weakly bisimilar to $N_2$. 
Section 6

Application to modular synthesis
(on the basis of \(\hat{N}\)-morphisms)
Synthesis

$$A = (S, E, T)$$
Elementary Transition Systems

$$N = (B, E, F)$$
Elementary Nets Systems

[ Nielsen, Rozenberg, Thiagarajan '92 ]
Modular synthesis

ETS and $\hat{G}$-morphisms

$A_I \xleftarrow{h_1} A_1$

$A_2 \xrightarrow{h_2} A_1$

$N(A_I) \xleftarrow{n_1} N(A_1) \xrightarrow{n_2} N(A_2)$

CG($N$) isomorphic to

[Bernardinello, Ferigato, Pomello 02]
Modular synthesis

ETS and $\hat{G}$-morphisms

\[
\begin{array}{ccl}
A_l & \xleftarrow{h_1} & A_1 \\
\uparrow h_2 & & \uparrow N(h_2) \\
A_2 & & N(A_2)
\end{array}
\]

ENS and $\hat{N}$-morphisms

\[
\begin{array}{ccl}
N(A_l) & \xleftarrow{N(h_1)} & N(A_1) \\
N(h_2) & & N(h_2)
\end{array}
\]

CG($N$) isomorphic to $A$
Modular synthesis

ETS and \( \hat{G} \)-morphisms

\[
\begin{array}{c}
A_I & \xleftarrow{h_1} & A_1 \\
\uparrow{h_2} & & \\
A_2 & \end{array}
\]

ENS and \( \hat{N} \)-morphisms

\[
\begin{array}{c}
N(A_I) & \xleftarrow{N(h_1)} & N(A_1) \\
\uparrow{N(h_2)} & & \uparrow{n_1} \\
N(A_2) & \xleftarrow{n_2} & N \\
\end{array}
\]

\( CG(N) \) isomorphic to

[Bernardinello, Ferigato, Pomello 02]
Modular synthesis

ETS and $\hat{G}$-morphisms

$A_i \xleftarrow{h_1} A_1 \xrightarrow{g_1} N(A_1)$

$N(A_2) \xleftarrow{n_2} N(A_1)$

ENS and $\hat{N}$-morphisms

$A_2 \xleftarrow{g_2} A \xrightarrow{h_2} N(A_1)$

$N(A_i) \xleftarrow{N(h_2)} N(A_1)$

$CG(N)$ is isomorphic to $A$

[Bernardinello, Ferigato, Pomello 02]
Section 7

A case study
Modeling and Analyzing
a Distributed Private Key Generation Protocol
Modeling Distributed Private Key Generation Protocol

Interaction of system components

Net representing PKG  Net representing client

Interface Net, $N_i$

- an abstract view of the system
- represents the communication between components
Interface Net, $N_i$

**Interface**
- an abstract view of the system
- represents the communication between components

**Properties**
- live
- reversible
- covered by sequential components
Net Representing PKG, $N_{PKG}$

Properties

- live
- reversible
- covered by sequential components
Net Representing the Client, $N_C$

Properties
- live
- reversible
- covered by sequential components
There is an $\alpha$-morphism both from $N_{PKG}$ to $N_I$ and from $N_C$ to $N_I$.

Additional requirements are satisfied.
• There is an $\alpha$-morphism both from $N_{PKG}$ to $N_I$ and from $N_C$ to $N_I$.

• Additional requirements are satisfied.

• Reflection of reachable markings property is held.

• Weakly bisimulation property is held.
  ▶ $N_{PKG}$ is weakly bisimilar to $N_I$.
  ▶ $N_C$ is weakly bisimilar to $N_I$. 
There is an $\alpha$-morphism both from $N_{PKG}$ to $N_I$ and from $N_C$ to $N_I$.

Additional requirements are satisfied.

Reflection of reachable markings property is held.

Weakly bisimulation property is held.

- $N_{PKG}$ is weakly bisimilar to $N_I$.
- $N_C$ is weakly bisimilar to $N_I$.
- Consequently, $N_{PKG} \langle N_I \rangle N_C$ is weakly bisimilar to $N_I$.
Ex:
A property to be analyzed: “Shares cannot be verified while distribution or extraction process is continuing”
Analysis on the Composed Net can be done directly on the Interface

even without computing the Composed Net

A property to be analyzed: "Shares cannot be verified while distribution or extraction process is continuing"

Corresponding CTL formulae

\[ \text{EXPATH EVENTUALLY shares verified AND key AND shares calculated} \]
Analysis on the Composed Net can be done directly on the Interface even without computing the Composed Net

A property to be analyzed: “Shares cannot be verified while distribution or extraction process is continuing”

Corresponding CTL formulae

EXPATH EVENTUALLY shares verified AND key AND shares calculated
Section 8

Remarks and Conclusions
Remarks and Conclusions

- \(\bar{N}\)-morphisms and the composition on \(\bar{N}\)-morphisms have been defined also for P/T nets

- an other notion of morphisms for marked graphs has been studied (paper just submitted to PNSE 2014)
Remarks and Conclusions

- N-morphisms and the composition on N-morphisms have been defined also for P/T nets
- An other notion of morphisms for marked graphs has been studied (paper just submitted to PNSE 2014)

Future?

- Define and study α-morphisms and the other just defined notion for more general classes (e.g.: P/T nets, high level nets, Coloured nets,..)
Remarks and Conclusions

- \(\hat{N}\)-morphisms and the composition on \(\hat{N}\)-morphisms have been defined also for P/T nets
- An other notion of morphisms for marked graphs has been studied (paper just submitted to PNSE 2014)

Future?

- Define and study \(\alpha\)-morphisms and the other just defined notion for more general classes (e.g.: P/T nets, high level nets, Coloured nets,..)
- Morphisms and compositionality on Petri Hypernets or on Nested nets


THANK YOU!

Спасибо большое!
THANK YOU!

Спасибо большое!

Arrivederci!...