A modular coloured Petri net model
for complex event processing

Ariane Piel

Directrice de thèse : Christine Choppy (LIPN, U. Paris 13 & CNRS)
Encadrants: Romain Kervarc et Patrice Carle (Onera, DCPS – TCS)

MeFoSyLoMa, le 4 octobre 2013
Systematic resort to simulation in the aerospace industry.

Study of all the characteristics of complex systems before even considering of undertaking them.

Generation of gigantic quantities of data: necessity to have an analysis assistance tool.

**ONERA** has a recognised expertise in distributed simulation:

- work on intention recognition initiated in 1998,
Main goals

- **Aim**: activity recognition in the framework of complex problems.
  - informatics security
  - supervision of medical environments

- Necessity of simulations since experiments may not be undertaken (because of criticality and cost for aerospace systems)
  - make use of the important quantity of data produced by the simulation in which numerous systems interact,
  - possibly, react to the data.
High criticality of the system → need for strong guarantees.

Very dynamic and highly complex data flow, be it between the agents of the system or between several systems.

**Aim**: overseeing the consistency of the system in case of failure(s)
An example: consistency of an unmanned aircraft system in case of breakdown(s)

- High criticality of the system → need for strong guarantees.
- Very dynamic and highly complex data flow, be it between the agents of the system or between several systems.
- **Aim**: overseeing the consistency of the system in case of failure(s)
An example: consistency of an unmanned aircraft system in case of breakdown(s)

- High criticality of the system → need for strong guarantees.
- Very dynamic and highly complex data flow, be it between the agents of the system or between several systems.
- **Aim**: overseeing the consistency of the system in case of failure(s)
High criticality of the system $\rightarrow$ need for strong guarantees.

Very dynamic and highly complex data flow, be it between the agents of the system or between several systems.

**Aim**: overseeing the consistency of the system in case of failure(s)

!! INCONSISTENT SITUATION !!
1 Chronicles: a behaviour detection formalism

2 Coloured Petri nets

3 A chronicle recognition model
Outline

1. Chronicles: a behaviour detection formalism
2. Coloured Petri nets
3. A chronicle recognition model
- a single event $A$
- a disjunction $C_1 \mid \mid C_2$ : at least either $C_1$ or $C_2$.
- a conjunction $C_1 \& C_2$ : both $C_1$ and $C_2$ in any order, possibly intertwined.
- a sequence $C_1 \cdot C_2$ : $C_1$ followed by $C_2$.
- an absence $(C_1) – [C_2]$ : $C_1$ without $C_2$ occurring during the recognition of $C_1$. 
Let $\mathcal{N}$ be a countable set of *single event names*. The set $X(\mathcal{N})$ of *chronicles* over $\mathcal{N}$ is defined inductively by these inference rules:

- **Name Rule**: $A \in \mathcal{N} \Rightarrow A \in X(\mathcal{N})$ (name)

- **Disjunction Rule**: $C_1, C_2 \in X(\mathcal{N}) \Rightarrow C_1 \parallel C_2 \in X(\mathcal{N})$ (disjunction)

- **Conjunction Rule**: $C_1, C_2 \in X(\mathcal{N}) \Rightarrow C_1 \& C_2 \in X(\mathcal{N})$ (conjunction)

- **Sequence Rule**: $C_1, C_2 \in X(\mathcal{N}) \Rightarrow C_1 \cdot C_2 \in X(\mathcal{N})$ (sequence)

- **Absence Rule**: $C_1, C_2 \in X(\mathcal{N}) \Rightarrow (C_1) - [C_2] \in X(\mathcal{N})$ (absence)
The *recognition set of C over flow ϕ until instant d*, denoted $R_C(\varphi, d)$, is defined by induction as follows:

- If $C = A \in N$, then
  $$R_A(\varphi, d) = \{(e, t) : \exists i \; \varphi(i) = (e, t) \land e = a \land t \leq d\}$$

- $R_{C_1 || C_2}(\varphi, d) = \{\langle r, \bot \rangle : r \in R_{C_1}(\varphi, d)\} \cup \{\langle \bot, r \rangle : r \in R_{C_2}(\varphi, d)\}$

- $R_{C_1 \& C_2}(\varphi, d) = \{\langle r_1, r_2 \rangle : r_1 \in R_{C_1}(\varphi, d) \land r_2 \in R_{C_2}(\varphi, d)\}$

- $R_{C_1 * C_2}(\varphi, d) = \{\langle r_1, r_2 \rangle : r_1 \in R_{C_1}(\varphi, D) \land r_2 \in R_{C_2}(\varphi, d) \land T_{\max}(r_1) < T_{\min}(r_2)\}$

- $R_{(C_1) - [C_2]}(\varphi, d) = \{r_1 : r_1 \in R_{C_1}(\varphi, d) \land \forall r_2 \in R_{C_2}(\varphi, d) \land (T_{\min}(r_1) > T_{\min}(r_2) \lor T_{\max}(r_1) \leq T_{\max}(r_2))\}$
Let $a, b, d$ and $e$ be events of $\mathcal{E}$ such that $\nu(a) = A$, $\nu(b) = B$, $\nu(d) = D$ et $\nu(e) = E$.

Consider chronicle $C = (A&B) - [D]$ and flow $\varphi = (a, e, b, a, d, b, a, a)$ where $\mathcal{O} \varphi = [1, 6]$.

$$R_C(\varphi, 8) = \{\langle 1, 3 \rangle, \langle 4, 3 \rangle, \langle 7, 6 \rangle, \langle 8, 6 \rangle\}$$
A coloured Petri net recognition model built by induction: for any given chronicle $C$, a Petri net $N(C)$ computing the recognitions of $C$

- modularity constraint
- determinism, in the sense that the nets always have to provide, given an event flow, the same recognitions
- maintaining concurrency
1 Chronicles : a behaviour detection formalism

2 Coloured Petri nets

3 A chronicle recognition model
Why coloured Petri nets?

- a concurrent framework
- a modular framework
  - to map the structure of the language
- a tool to visualise executions and check properties
Coloured Petri nets

CPN definition

A non-hierarchical CPN is a tuple \((P, T, A, \mathcal{B}, V, C, G, EX, I)\) :

1. \(P\) is a finite set of places.
2. \(T\) is a finite set of transitions such that \(P \cap T = \emptyset\).
3. \(A \subseteq P \times T \cup T \times P\) is a set of directed arcs.
4. \(\mathcal{B}\) is a finite set of non empty colour sets (types).
5. \(V\) is a finite set of typed variables such that : \(\forall v \in V\) \(Type[v] \in \mathcal{B}\).
6. \(C : P \rightarrow \mathcal{B}\) is a colour set function assigning a colour set to places.
7. \(G : T \rightarrow \text{Expr}_{\Sigma(V)}\) is a guard function.
8. \(EX : A \rightarrow \text{Expr}_{\Sigma(V)}\) is an arc expression function.
9. \(I : A \rightarrow \text{Expr}_{\Sigma(V)}\) is an initialisation function.

A non-hierarchical CPN is a tuple \((P, T, A, B, V, C, G, EX, I, B)\):

1. \(P\) is a finite set of places.
2. \(T\) is a finite set of transitions such that \(P \cap T = \emptyset\).
3. \(A \subseteq P \times T \cup T \times P\) is a set of directed arcs.
4. \(B\) is a finite set of non empty colour sets (types).
5. \(V\) is a finite set of typed variables such that: \(\forall v \in V\ \ Type[v] \in B\).
6. \(C : P \rightarrow B\) is a colour set function assigning a colour set to places.
7. \(G : T \rightarrow \mathit{Expr}_{\Sigma (V)}\) is a guard function.
8. \(EX : A \rightarrow \mathit{Expr}_{\Sigma (V)}\) is an arc expression function.
9. \(I : A \rightarrow \mathit{Expr}_{\Sigma (V)}\) is an initialisation function.
10. \(B \subseteq P \times T\) is a set of directed inhibitor arcs.
Coloured Petri nets

A modularity mechanism: place fusion

- strong constraint: requirement that the model be compositional so as to map the inductive structure of the chronicle language.

- In [CP92], definition of Modular CPN $MCPN = (S, PF, TF)$.


- Problematics of the definition of place fusion:
  - compositional model $\rightarrow$ fusion of MCPN instead of CPN
  - CPN Tools functionality: change of initial markings and initial types after a place fusion.

- Accordingly, modification of [CP92] to define MCPN, MCPN fusion, and the resulting CPN.
A chronicle recognition model

Outline

1 Chronicles: a behaviour detection formalism

2 Coloured Petri nets

3 A chronicle recognition model
A chronicle recognition model

Design stages

1st model:

- modular model
- one token in each place containing lists of recognitions
- non-deterministic: necessity to define a transition firing strategy
- non-concurrent
- no event flow management


A chronicle recognition model

Design stages

2nd model: transition to multi-token nets

- modular model
- one token for each recognition
- first control structures
- beginning of some concurrency
- no event flow management
3rd model: adding a control structure for event flow management

- modular model
- one token for each recognition
- deterministic model
- concurrent model
- event flow management

A chronicle recognition model

Structure of net recognising chronicle $A$ (including the event counter)

event counter (CPT)
A chronicle recognition model

Net recognising chronicle A (including the event counter)
A chronicle recognition model

Token splitter
Nets will be built in 2 steps:

1. main mechanism of the recognition process $N'(C)$
2. fusion with the event counter $CPT$ to build $N(C)$

$$N(C) = Fusion \left( \{ N'(C), CPT \}, \{ (Go(CPT), \{ Go(CPT), Go(C) \} ), \right.$$  
$$\left. (Present(CPT), \{ Present(CPT), Present(C) \} ), \right.$$  
$$\left. (End(CPT), End(CPT), End(C) \} \right) \}$$
A chronicle recognition model

Sequence and disjunction on chronicle $A \parallel (B \ A)$
A chronicle recognition model

Absence on chronicle \((A\ B) – [D]\)
**A chronicle recognition model**

**State Space of** \( N(A \parallel (B \ A)) \) **on event flow** \( \varphi = (b, a, a) \)

- regardless of the order in which the enabled transitions are fired, always the same recognitions after each event
- concurrency retained
A chronicle recognition model

Conclusion and perspectives

- a formal framework for behaviour recognition using CPN with inhibitor arcs and place fusion
- strong constraints on the model:
  - compositionality,
  - determinism,
  - concurrency

- extend the construction of the nets to other constructs of the chronicle language (continuous time constraints, dealing with event enriched with attributes, ...)
- prove the adequacy of the Petri net model with the set semantics of chronicles
- put to use our Petri net behaviour recognition model for applications in the aerospace industry