Supervision of Concurrent Systems using (High-level/Time) Petri Net Unfoldings

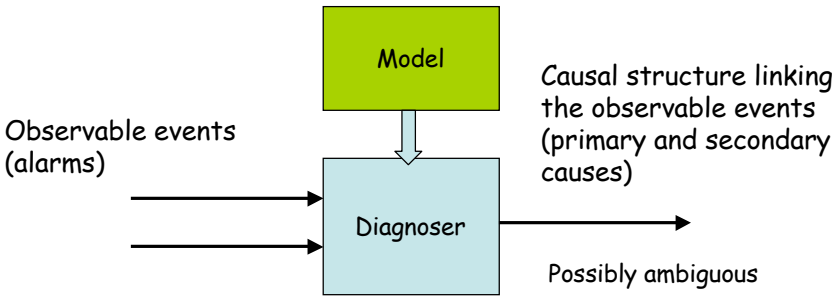
Claude JARD

Ecole Normale Supérieure de Cachan
IRISA/Distribcom research group

Rennes, France.

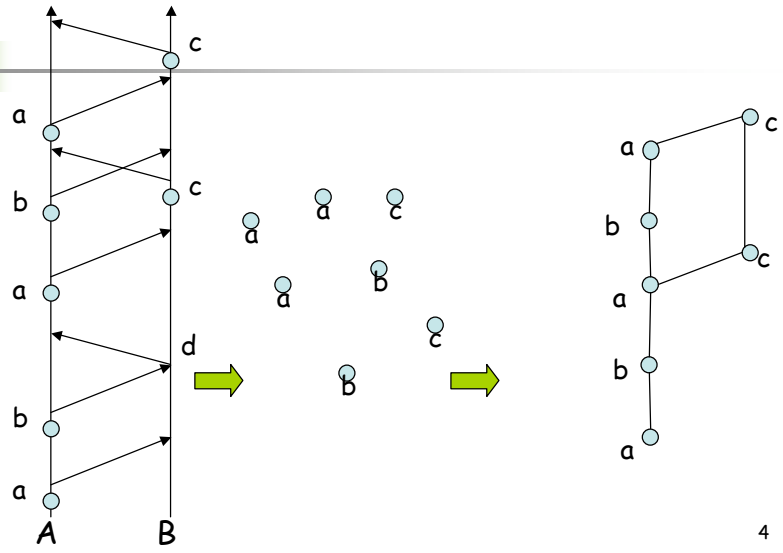
- Supervision is one of the key-challenge in mastering telecommunication networks
- Concurrency aspects must be specifically considered, especially for the large-scale
- We chose a model-based approach, which can be well formalized using Petri nets (and UML-embedded in our industrial collaborations)
- We use the notion of unfolding to capture causal dependencies and conflicts between the observable events
- Extensions are needed to deal with more complex systems (symbolic, timed or dynamic models)

Model-based diagnosis



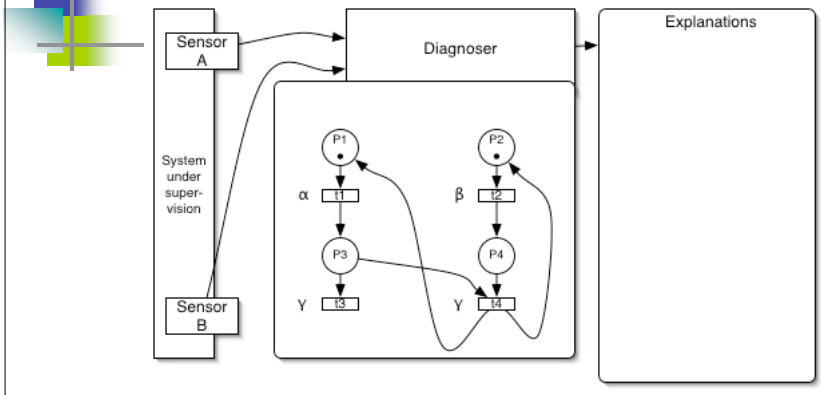
3

Causal inference

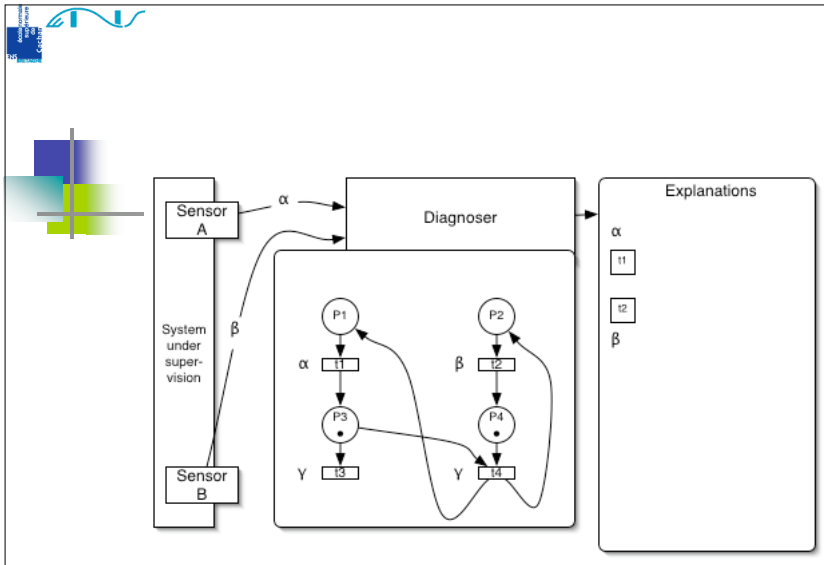


4

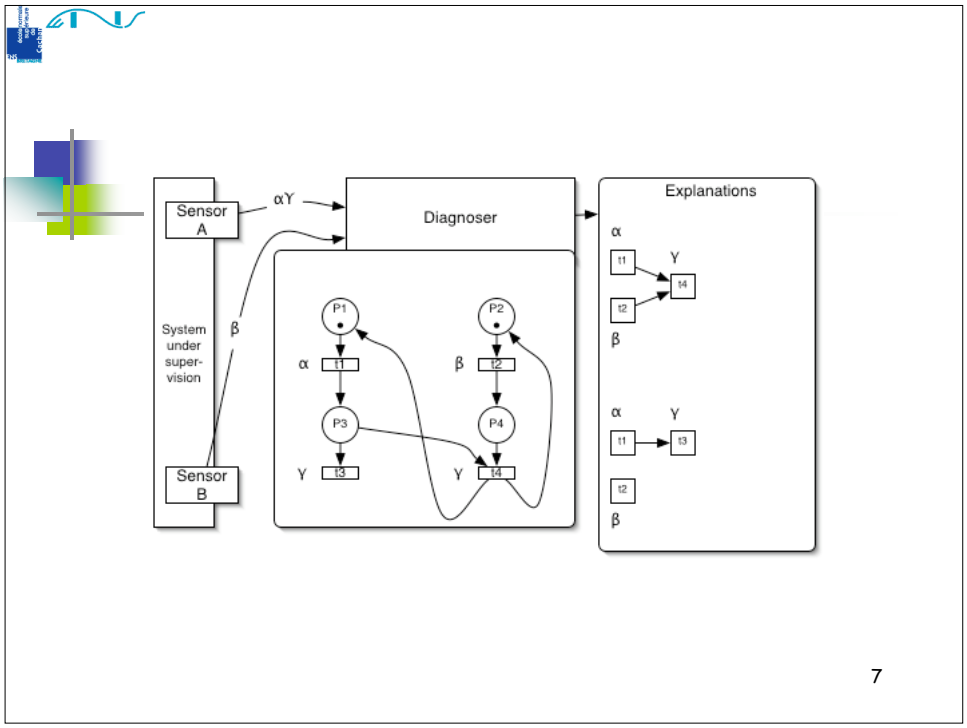
A toy example



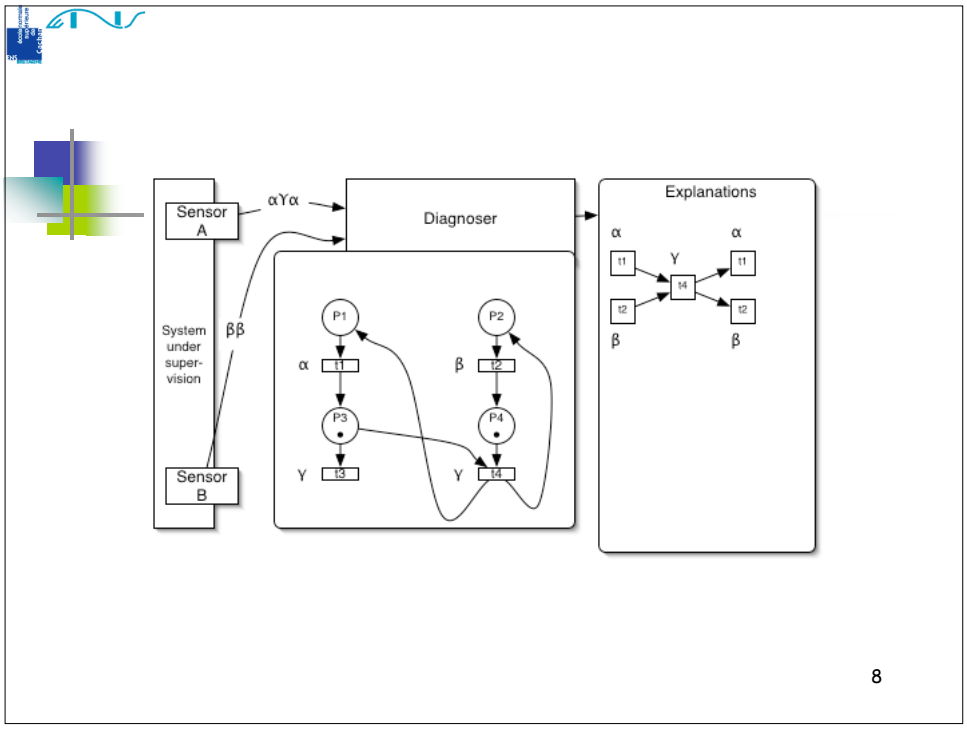
5



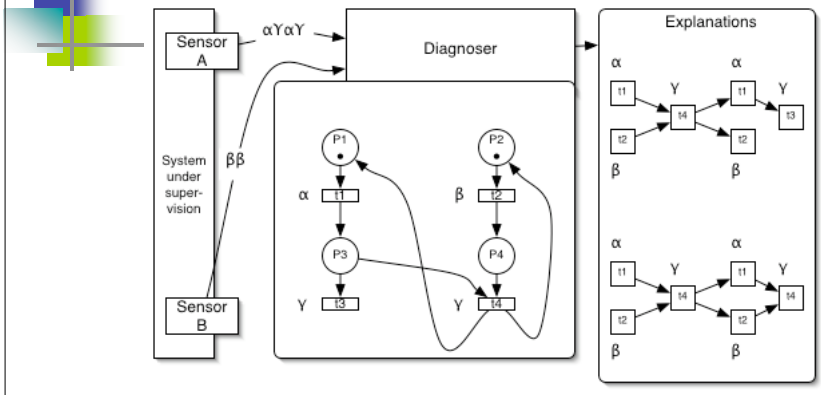
6


Sensor A
αγ
Diagnoser
Explanations
System under super-vision
β
P1
P2
α
t1
β
t2
P3
P4
γ
t3
γ
t4
Sensor B
α
t1
γ
t4
t2
β
α
t1
γ
t3
t2
β

7


Sensor A
αγα
Diagnoser
Explanations
System under super-vision
ββ
P1
P2
α
t1
β
t2
P3
P4
γ
t3
γ
t4
Sensor B
α
α
t1
γ
t4
t1
t2
t2
β
β

8



9

Petri Nets

N=(P,T,->,M_0,Σ ,λ)

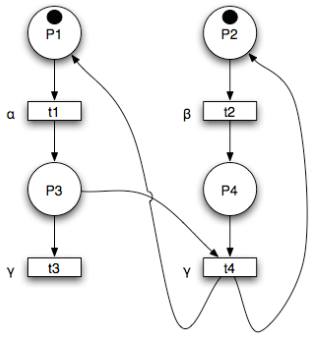
- P and T disjoint finite sets of places and transitions, -> flow relation in (PxT ∪ TxP)
- λ : T -> Σ labeling of transitions (alarm names)
- ≤ et < are the transitive closures of ->
- For x ∈ P∪T, preset °x = {y | y -> x},
  postset x° = {y | x -> y}
- A marking is a multiset M : P -> {0,1,2, …}
- M_0 : initial marking
- t∈T, t is firable in the marking M iff °t ≤ M
- If t fires, M := (M - °t) + t°

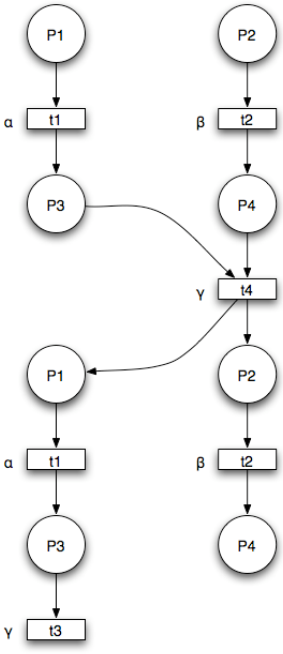
10

Processes

- Events are transitions occurrences : E ($\perp \in E$)
- $e \in E$, $e = ({}^\circ e, \tau_e)$, $\tau_e \in T$, ${}^\circ e = \{({}^\circ b, place(b))\}$
- ${}^\circ b$ is the event that has created a token in place(b)
- b° is the event that consumes a token in place(b)
- Place(B) = {| place(b), $b \in B$ |}
- $\uparrow E = \cup_{e \in E}\, e^\circ \setminus \cup_{e \in E}\, {}^\circ e$
- The set X of all processes is defined inductively by:
  - $\emptyset \in X$
  - For all process $E \in X$, for all transition t, and for all set $B \subseteq \uparrow E$ such that $Place(B) = {}^\circ t$, $E \cup \{e\} \in X$, where $e = (B, t)$

Example of process



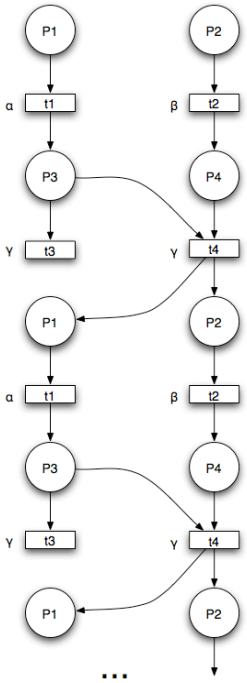


Unfolding: superimposition of all processes

Consider the union U of processes

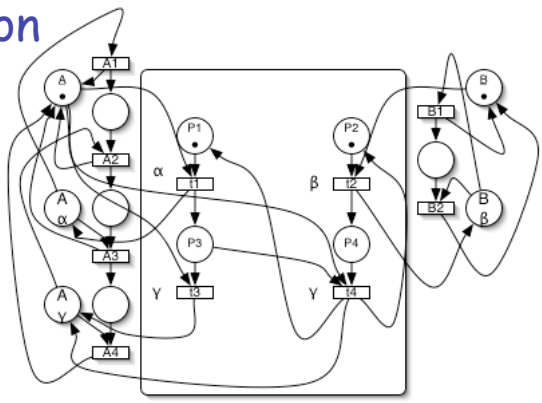
Process extraction:

- $\lceil e \rceil = \{e' | e' \leq e\}$ (The unfolding is itself a PN)
- $E \subseteq U$ is a process iff:
  - $\lceil E \rceil = E$ (E is causally closed)
  - $\forall e,e' \in E\ {}^\circ e \cap e'^\circ = \varnothing$ (E is conflict free)

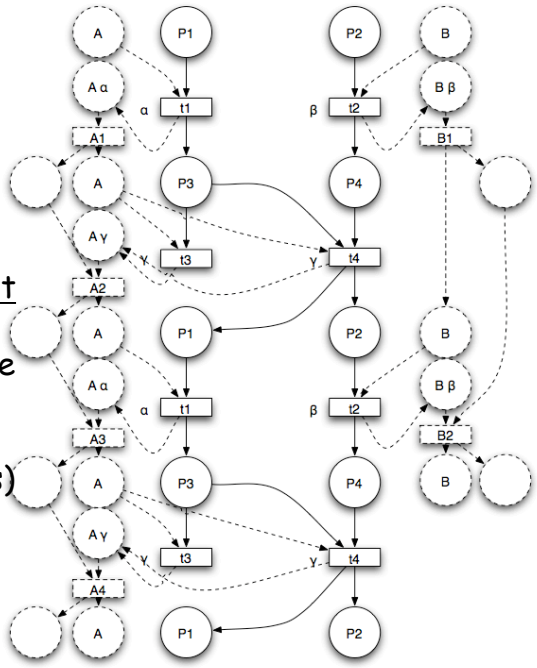


On-line construction, guided by the observation

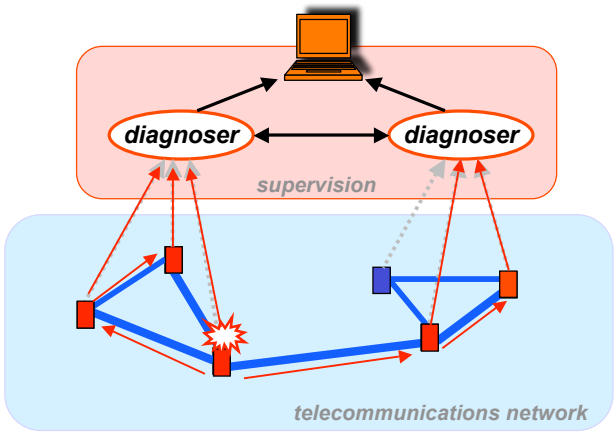
- Causal observation (sequential activities of sensors are observed consistently)
- Sensors can be modeled with PN



- Unfolding of the product (observations X model)
- The explanations are the processes (projected) of the (finite) unfolding (underlying partial orders)



The real life with Alcatel…



- Fault propagation – causality
- Alarm interleaving – concurrency
- Distributed processing

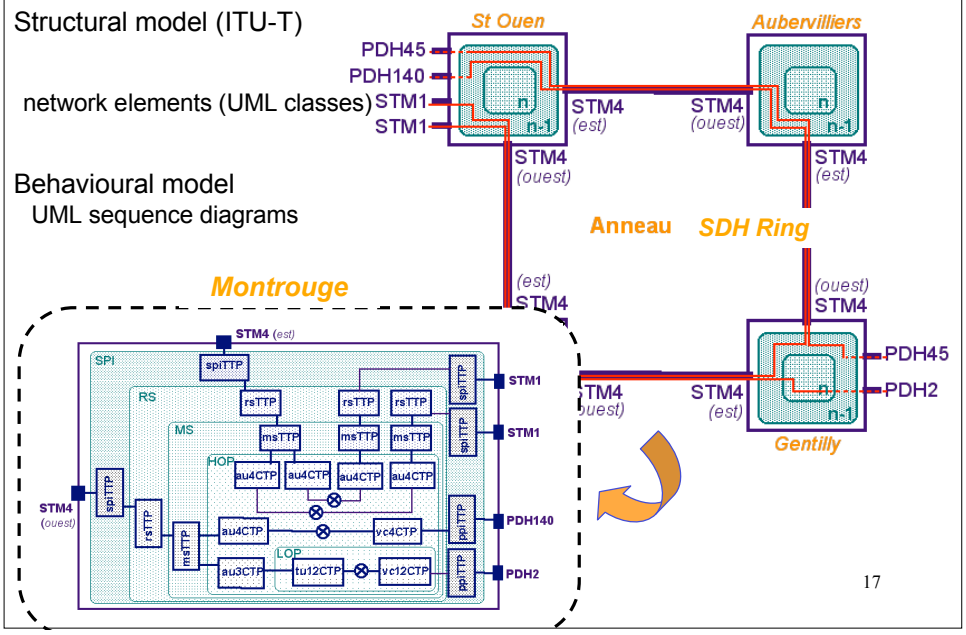
16

Structural model (ITU-T)

network elements (UML classes)

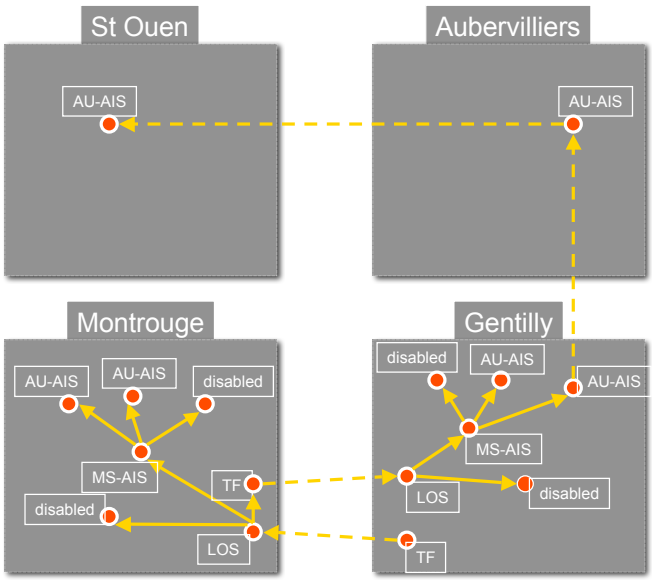
Behavioural model

UML sequence diagrams



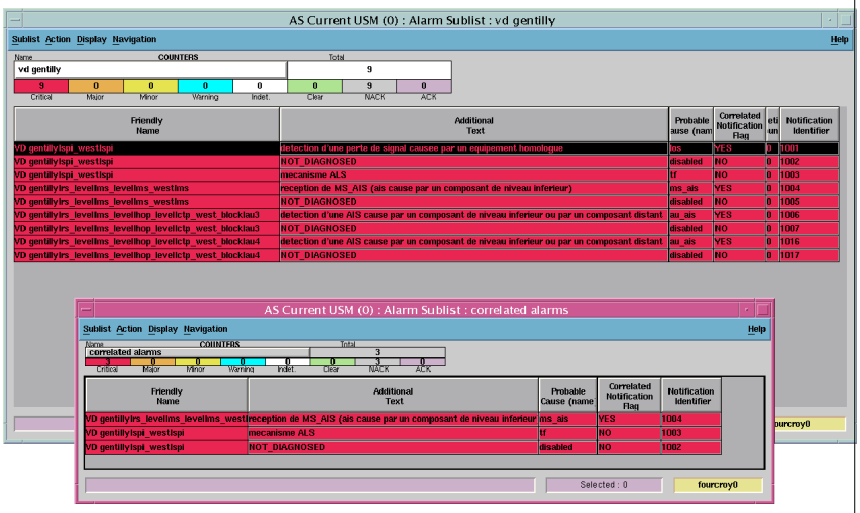
17

Inferring causalities from local observations



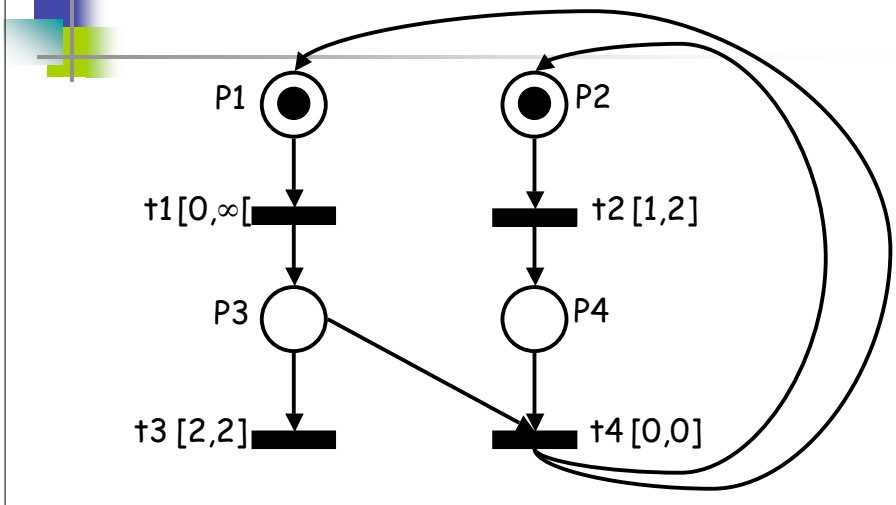
18

Integration with their tool



19

Extension to Safe Time Petri nets



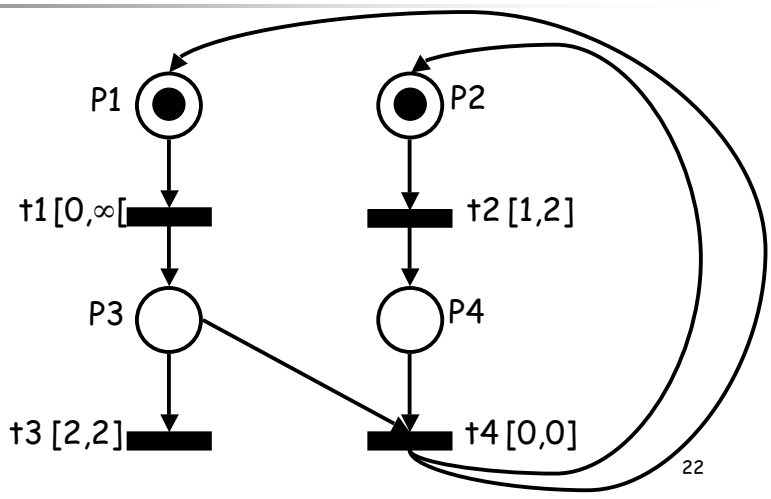
20

Dynamics

- Clock-function I(t) = [↓I(t), ↑I(t)] (non negative rationals); $I_s(t)$ is the given static interval
- $I(t)\backslash\Theta = [\max(0,\downarrow I(t)-\Theta), \max(0,\uparrow I(t)-\Theta)]$
- $(M,I) \rightarrow_{t@\Theta} (M',I')$ iff
  - $(M \geq {}^\circ t) \wedge \downarrow I(t) \leq \Theta \wedge \forall\, t' : (M \geq {}^\circ t \Rightarrow \Theta \leq \uparrow I(t'))$
  - $M' = (M \backslash {}^\circ t) \cup t^\circ$
  - $\forall\, t' : (M' \geq {}^\circ t' \Rightarrow$
    $I'(t') = \text{if } (t' \neq t \wedge M \backslash {}^\circ t \geq {}^\circ t') \text{ then } I(t')\backslash\Theta \text{ else } I_s(t')$

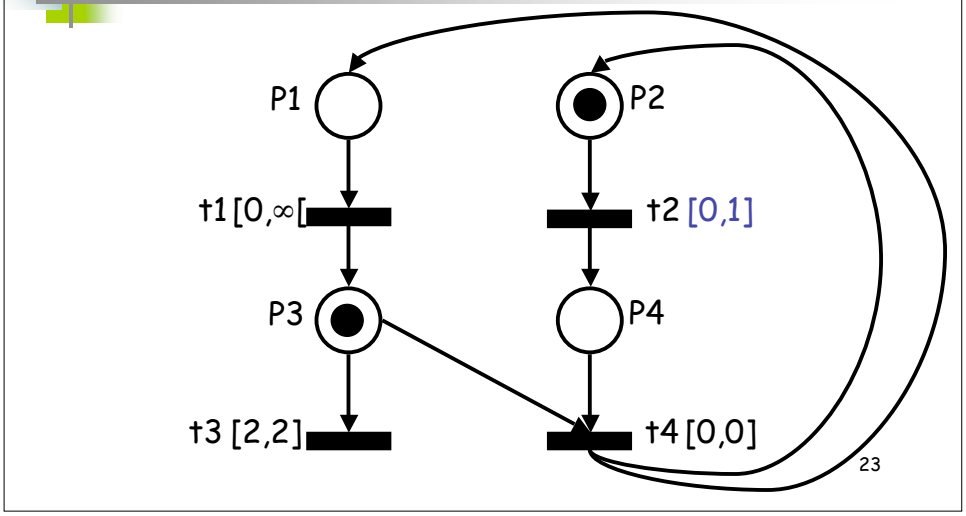
21

$(P_1P_2, t_1^{[0,\infty[} t_2^{[1,2]} t_3^{[2,2]} t_4^{[0,0]})$



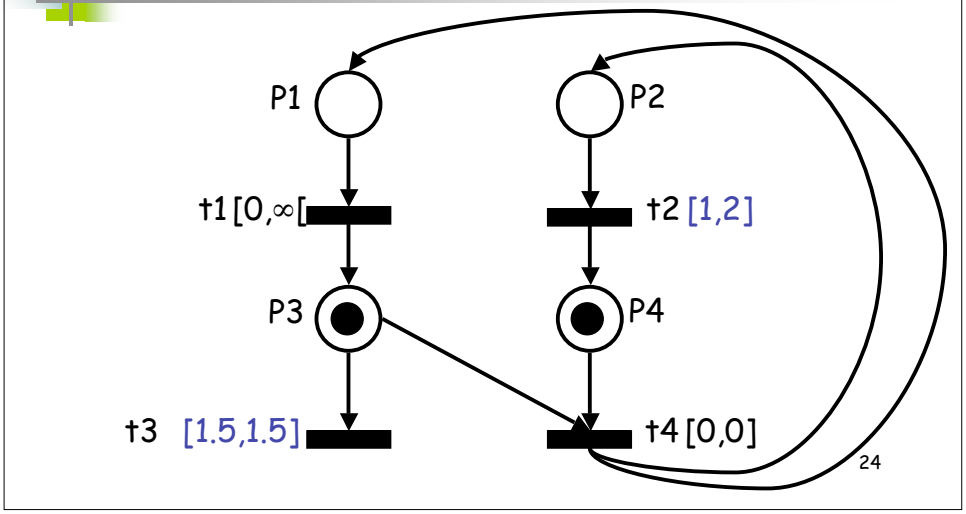
22

$(P_1P_2, t_1^{[0,\infty[}t_2^{[1,2]}t_3^{[2,2]}t_4^{[0,0]})\text{-}t_1@1\text{->}(P_2P_3, t_1^{[0,\infty[}t_2^{[0,1]}t_3^{[2,2]}t_4^{[0,0]})$



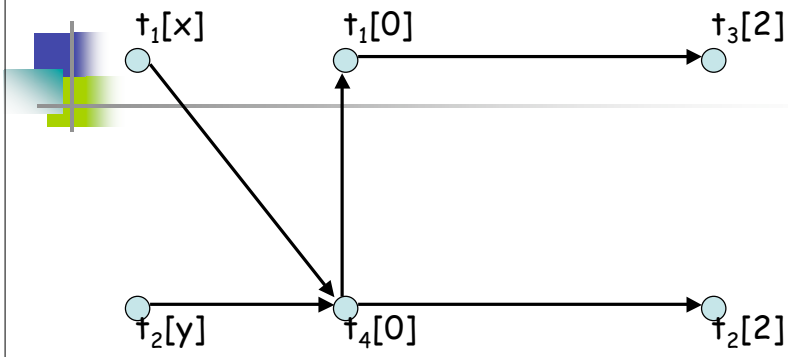
23

$(P_1P_2, t_1^{[0,\infty[}t_2^{[1,2]}t_3^{[2,2]}t_4^{[0,0]})\text{-}t_1@1\text{->}(P_2P_3, t_1^{[0,\infty[}t_2^{[0,1]}t_3^{[2,2]}t_4^{[0,0]})$
$\text{-}t_2@0.5\text{->}(P_3P_4, t_1^{[0,\infty[}t_2^{[1,2]}t_3^{[1.5,1.5]}t_4^{[0,0]})$



24

Time trajectories

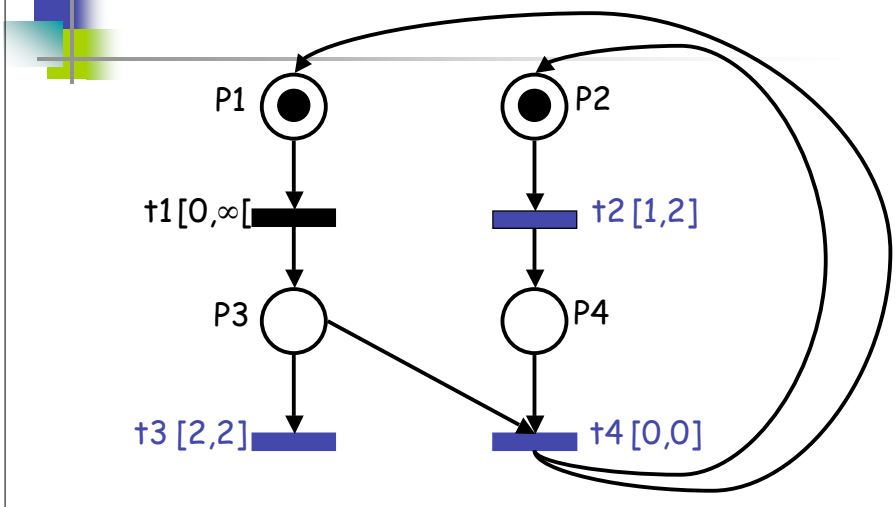


1≤y≤2

are constrained trajectories of the underlying PN

25

Difficulty : non locality (confusion)



26

Principle: work in the general framework of high-level PN with parameters

- Introduction of the complementary conditions (places) to test the emptyness in order to know that a given transition is no more fireable
- Also implies the introduction of read arcs in order to avoid to generate conflicts when two events (transitions) only read the same information (do not sequentialize)
- Introduction of formal parameters to represent:
  - the possible delays before the fire of each event (Θ_t) (x,y, … in our example)
  - the date of removal of tokens in each complementary conditions (m_p)
- Each condition carries its date of birth
- Each event carries a constraint on parameters

27

Symbolic unfolding (enabling condition)

- Each condition *c* carries its date of birth, denoted *d(c)*
- Let h the mapping from the conditions to P x {empty,full)
- An event *e* corresponding to a transition *t* can be put in the unfolding on the set *C* of consumed conditions and *R* of read conditions iff:

1. $C \cap R = \emptyset$
2. $C \cup R$ is a co-set
3. $h(C) = \{(p,full) \mid p \in {}^\circ t\} \cup \{(p,empty) \mid p \in t^\circ\}$
4. Let $D = \{t' \in T \mid t' \# t\} \cup \{t' \in T \mid \exists p \in t'^\circ\ (p,empty) \in h(R)\}$
   It is required that transitions of D do not fire before *t*
   $\forall t' \in D\ \exists p \in {}^\circ t' : (p,full) \in h(C \cup R) \vee (p,empty) \in h(C \cup R)$
5. The predicate associated with *e* is satisfiable (see next slide)

28

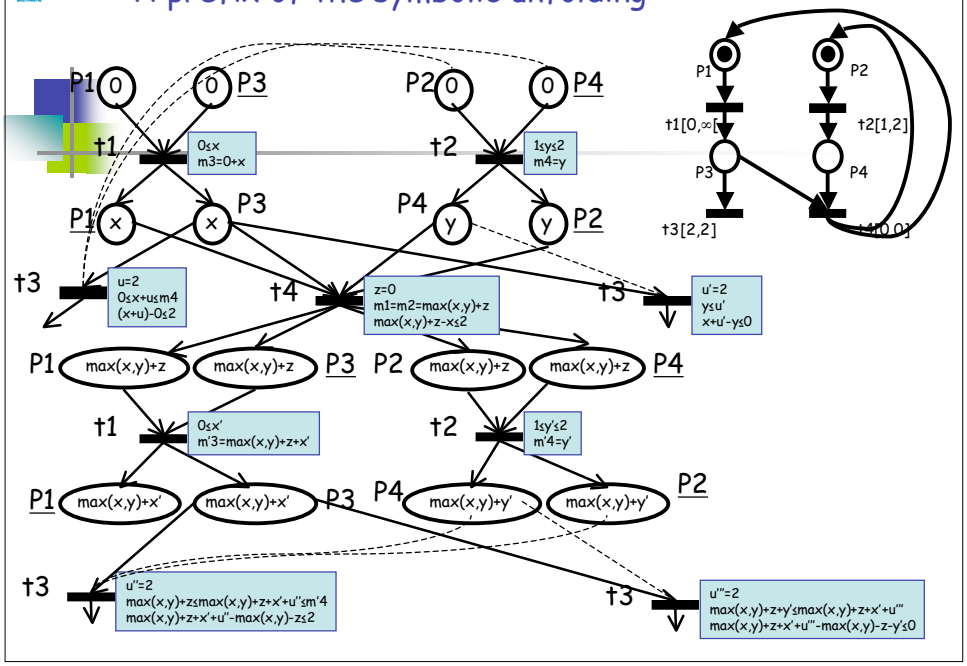
Symbolic unfolding (event predicate)

- Let Θ the delay before firing t
- $d(e) = \max_{p \in {}^\circ t}(d(h^{-1}(p,full))) + \Theta$ is the firing date of e
- The predicate associated with e is:

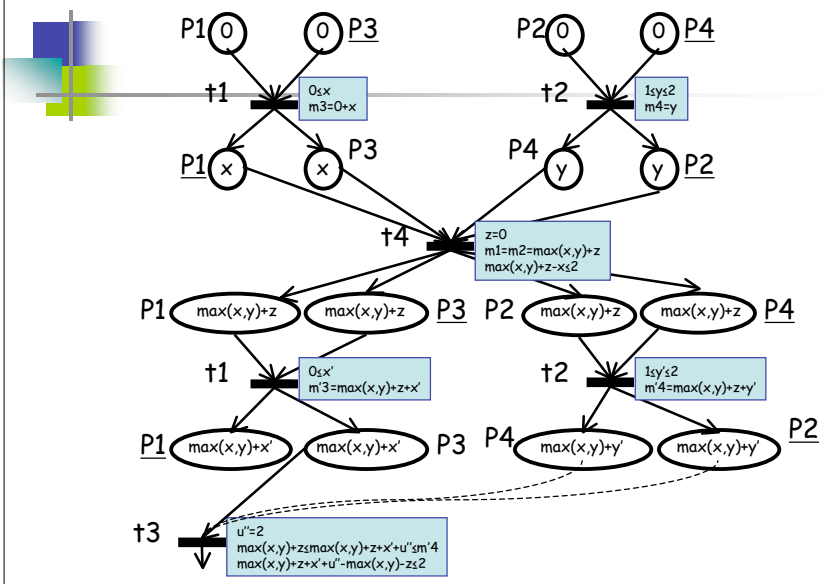
1. $dmin(t) \leq \Theta \leq dmax(t)$
2. $\forall p \in t^\circ : m_{h-1(p,empty)} = d$ (the output places of t are filled at date d)
3. $\forall c \in R : d(c) \leq d$ (the conditions that are read are born before d)
4. $\forall t' \in D, \forall p \in {}^\circ t' : d - d(h^{-1}(p,full)) \leq dmax(t')$
   (the transitions of D have not fired)
5. $\forall (p,empty) \in h(R) : d \leq m_{h-1(p,empty)}$ ()

29

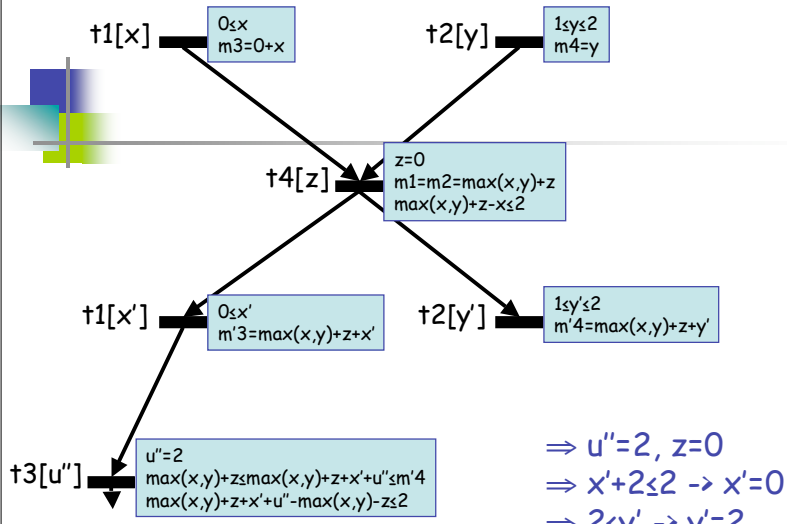
A prefix of the symbolic unfolding



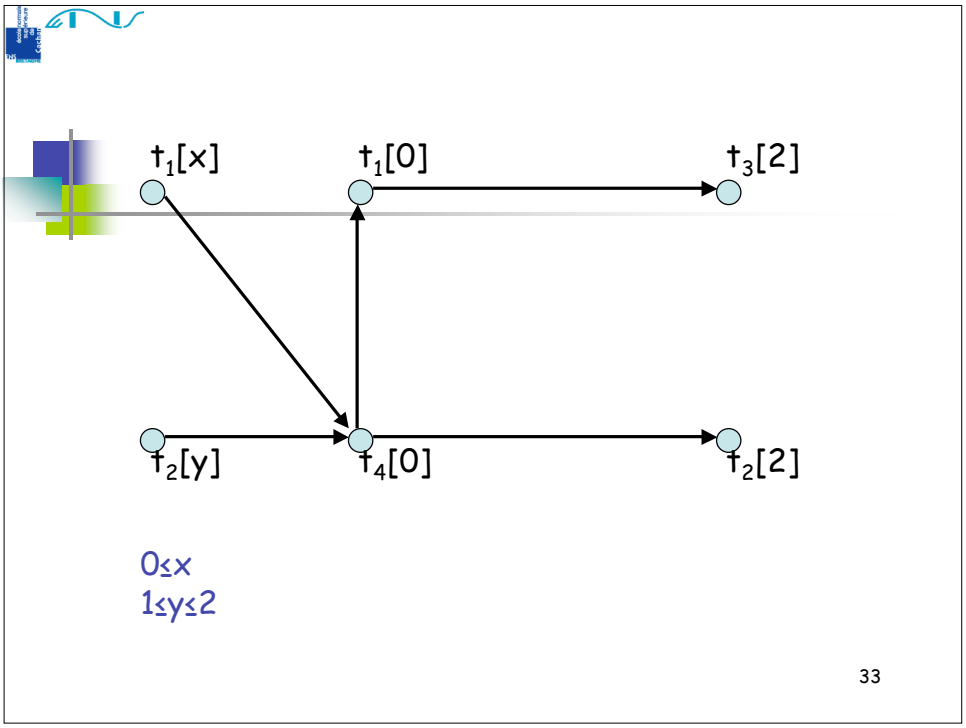
A possible configuration



The underlying time trajectory

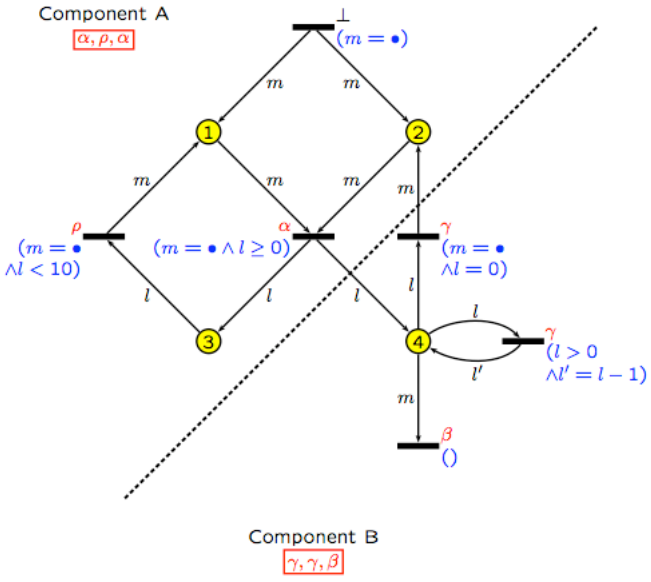


⇒ u''=2, z=0
⇒ x'+2≤2 -> x'=0
⇒ 2≤y' -> y'=2



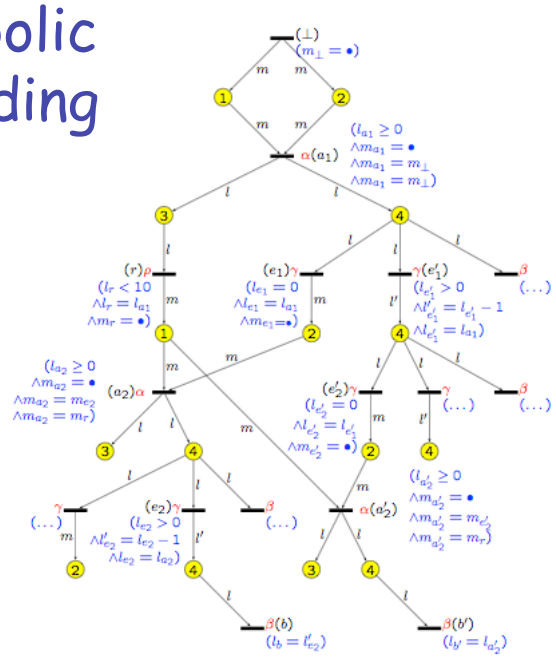
33

High-level models

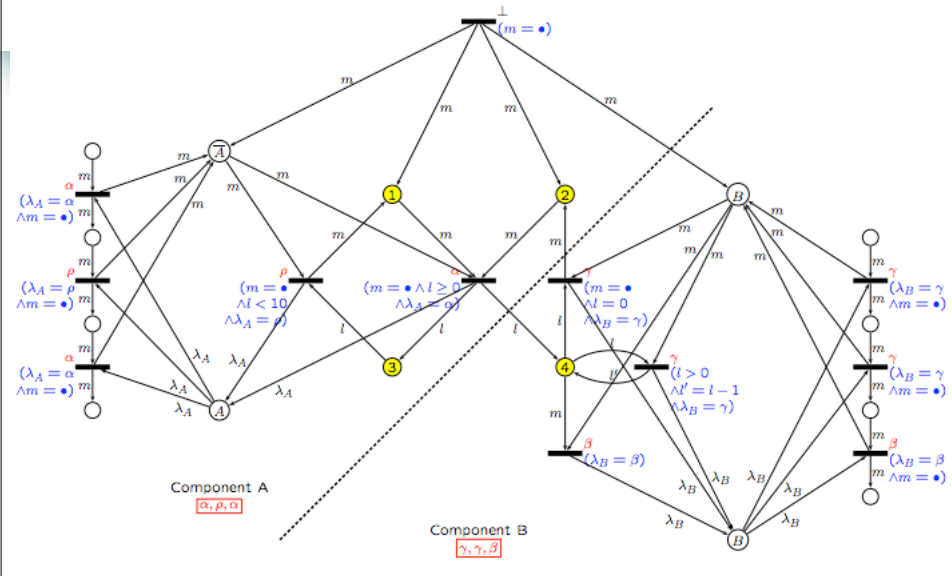


34

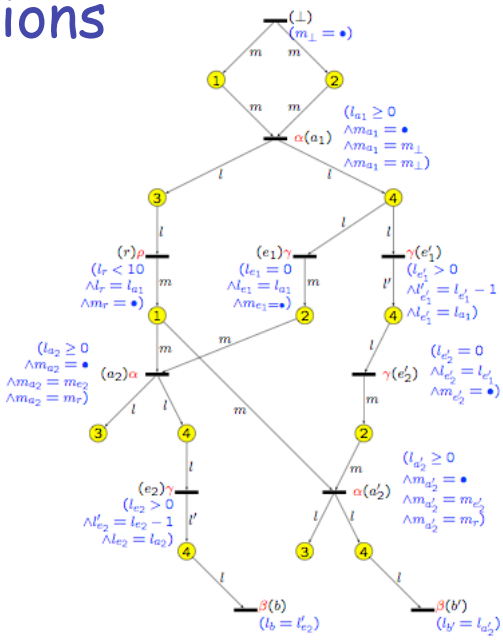
Symbolic unfolding



Model with observations



Explanations



Summary

- A model-based supervision approach, dedicated to concurrent systems
- Formally presented using Petri nets
- Extensible to more expressive models (symbolic supervision)
- Other works: dynamic models, stochastic models, networks of automata…
- Current interest on model robustness, observability and control

38