# Decentralised diagnosis of discrete-event systems: application to telecommunication network

Yannick Pencolé
CSL, The Australian National University
Yannick.Pencole@anu.edu.au

in collaboration with M.-O. Cordier and L. Rozé

# Telecommunication network

- Set of connected components
  - Distributed system
  - Metropolitan/Wide Area Network
- Purpose: transmission of data between clients (companies)
- Network management: providing a good quality of services
  - "Using all the network resource with a minimal cost"
  - Traffic management, failure management

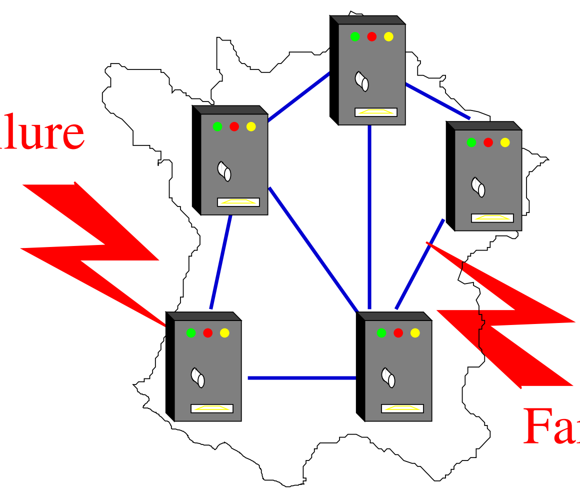
# Network monitoring

Supervision center

Purpose:
Identify the problems
that could have occurred
which explain the received
alarms

Alarms

Failure



Failure

Detection,
Localisation,
Identification,
Propagation

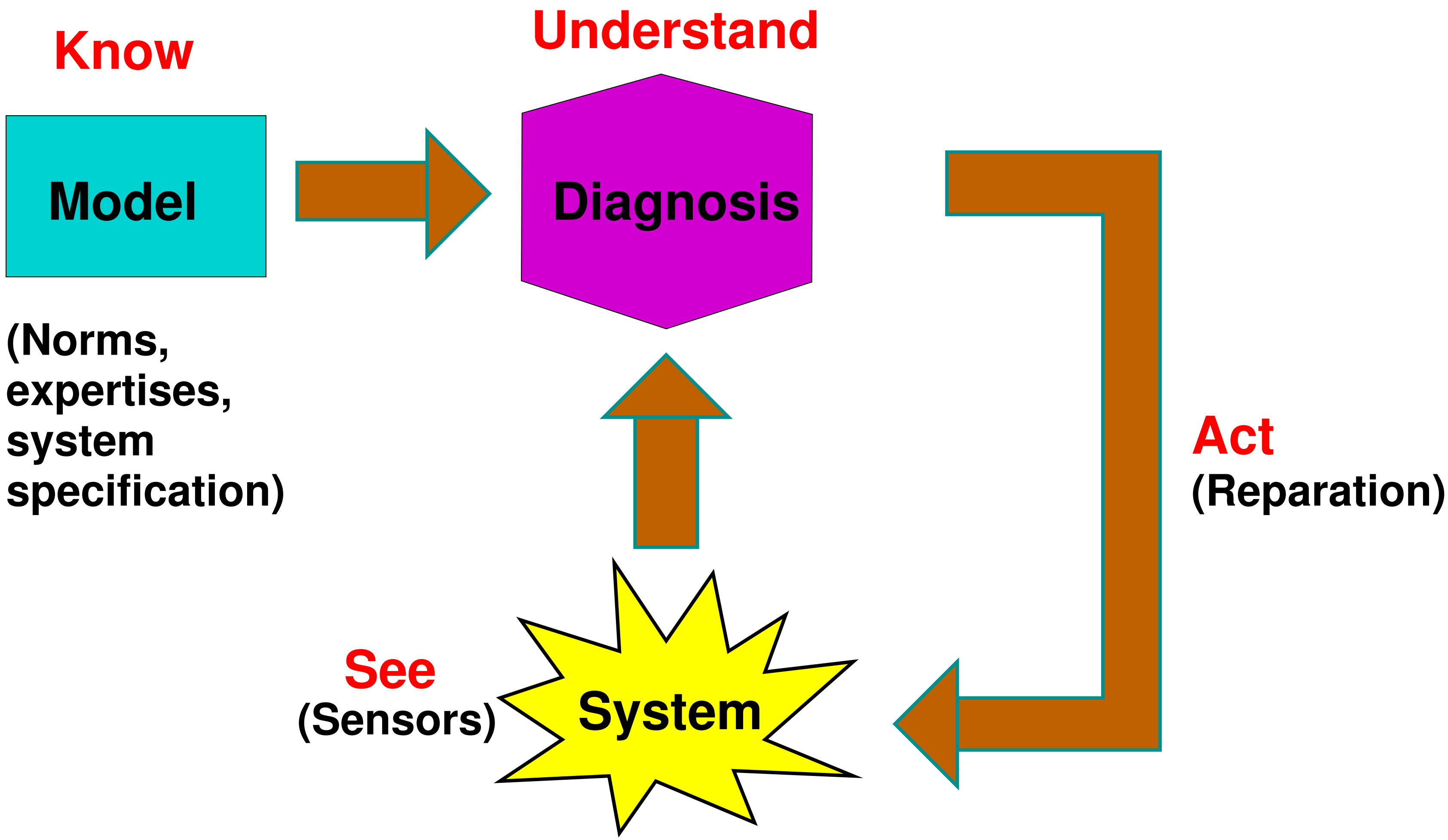
# Needs for monitoring

- Supervised network: large scale system
  - Very important number of received alarms per day (several thousands)
  - Supervisor: human agents
    - Analysis of the received alarms: complex problem, in particular if we need to determine the problems quickly
- An automatic system to help in the alarm interpretation is necessary
- Existing systems [Sloman86][Jakobson93]
  - Expert systems, correlation alarm systems
    - Problem: evolutive system

# Context and purposes

- To propose a failure diagnosis system
  - Taking into account the evolutivity of the supervised system
  - Producing complete and concise diagnoses
  - Online diagnosis approach: need of efficiency
- Our proposal: decentralised approach
- Context: MAGDA project
  - Academic partners: IRISA, LIPN
  - Industrial partners: Alcatel, France Telecom, Ilog

# Diagnosis: principles



# KNOW: Model

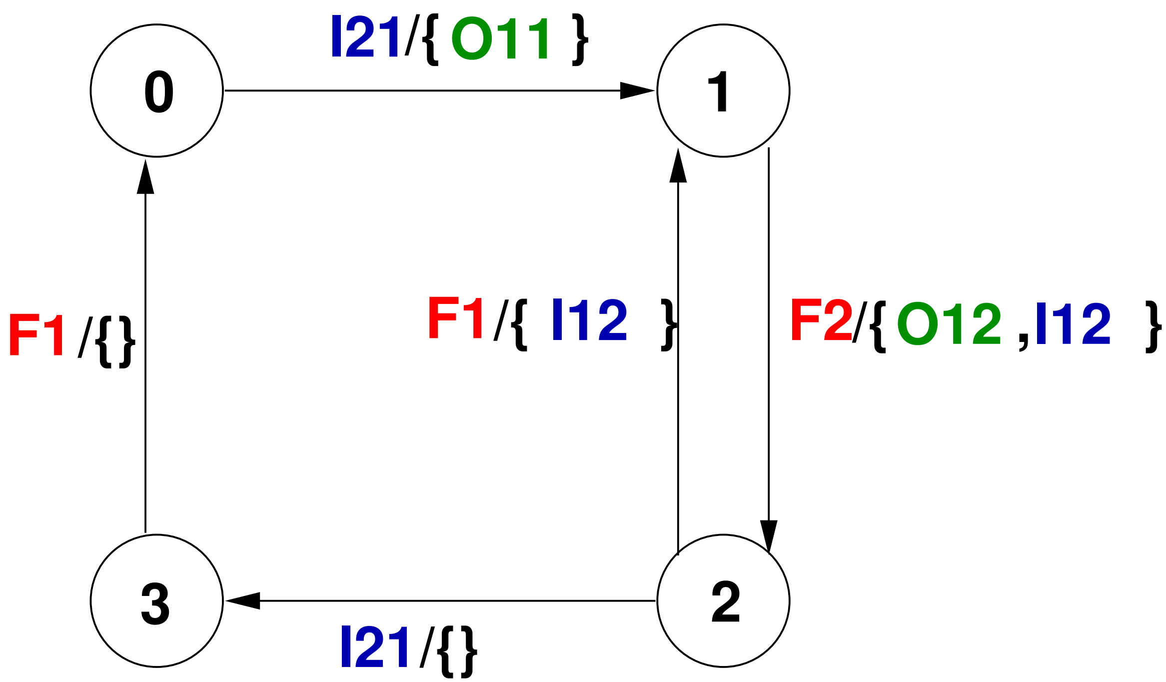
- Model of discrete-event systems
  - *Failure*: occurrence of an event which can change the state of a component
  - *Interaction* between components: message exchanges (emission/reception)
  - *Alarm*: emission of an observable event by a component
- Behaviour of the system
  - Nominal behaviour, Faulty behaviour
- Used formalism:
  - Set of communicating automata

# Model of component

$$\Gamma_i = (Q_i, Exo_i \oplus Rcv_i, Obs_i \oplus Emit_i, T_i)$$

- $Q_i$ finite set of states, modes of behaviour (faulty or not)
- Reception events
  - $Exo_i$ exogenous events: failures, actions from the environment
  - $Rcv_i$ internal events: reception of messages from other components (event propagation)
- Emission events
  - $Emit_i$ internal events: emission of messages to other components (event propagation)
  - $Obs_i$ observable events: emission of alarms to the supervisor
- $T_i \subseteq Q_i \times Exo_i \oplus Rcv_i \times 2^{Obs_i \oplus Emit_i} \times Q_i$ set of transitions

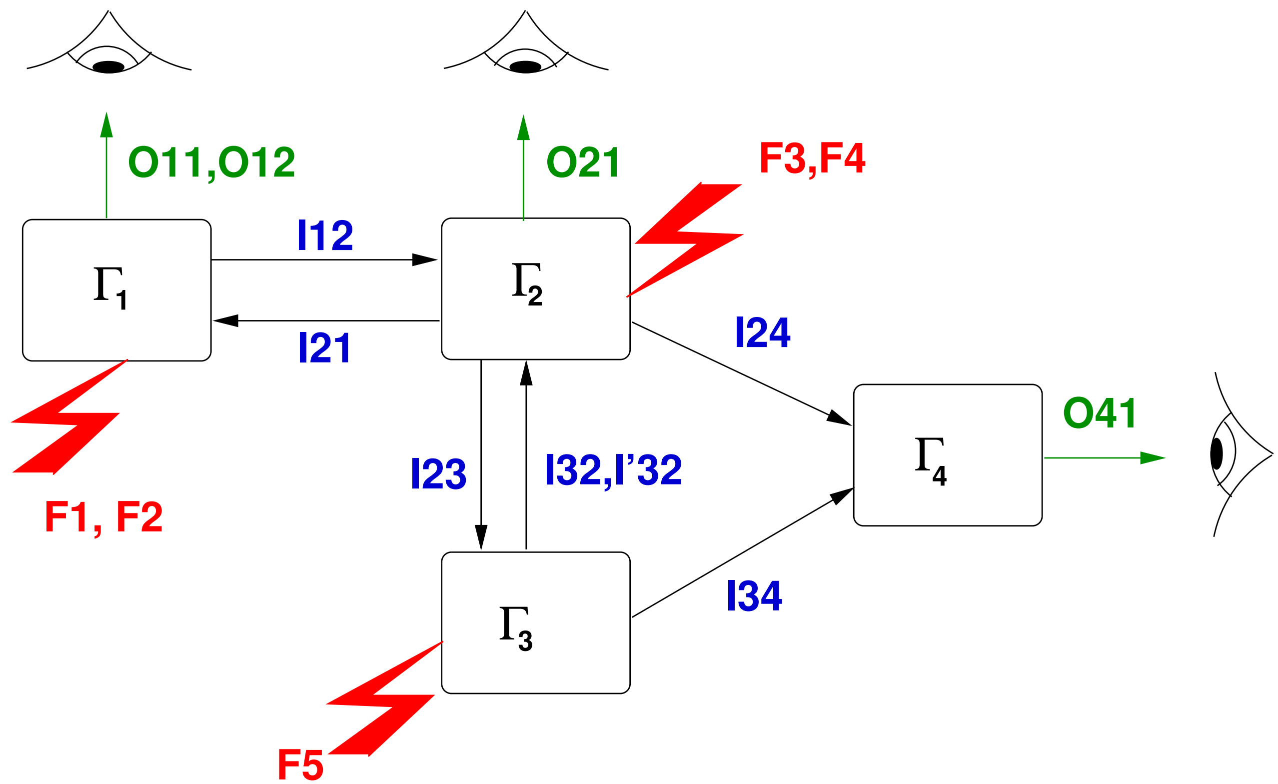
# Model of component: example




- $Exo_i$: **F1 F2**, $Rcv_i$: **I21**
- $Emit_i$: **I12**, $Obs_i$: **O11 O12**

# Model of the system

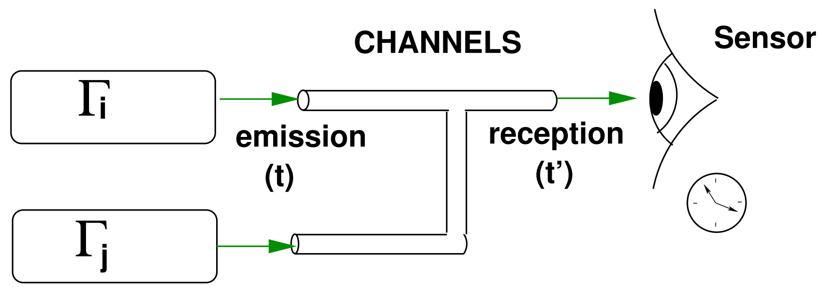
Set of components: $\Gamma \triangleq \{\Gamma_1, \ldots, \Gamma_n\}$




Global behaviour obtained by synchronised product (synchronisation on internal events): $\|\Gamma\| = \prod_{i\in 1}^{n} \Gamma_i$

# SEE: Observations

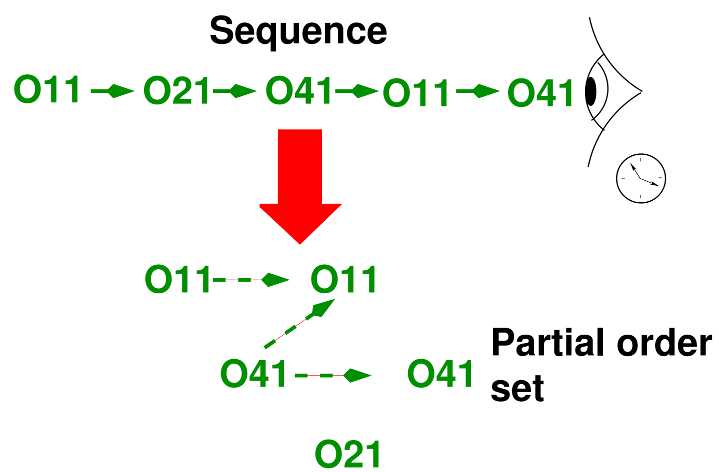
- Set of sensors
  - Observation channels




  - Different propagation delays
    - Instantaneous
    - With a known maximum delay $D$
- Observation: reception of a message from a component by a sensor
- On a sensor: order of reception $\neq$ order of emission!
- 2 sensors may not have synchronised clocks!

# Observations: partial order

- $\mathcal{O}$: set of observations (message with a date of reception)
- $\preceq$: partial order relation on the observations
  - based on the *observability of the system*
    - Number of sensors (synchronised clocks?)
    - Characteristics of the channels
      - Instantaneous ? FIFO ? propagation delay ?



# UNDERSTAND: diagnosis

- Purpose: to explain the observations by the occurrence of failures (permanent, intermittent)
  - Given the model $\Gamma$, given the observations $\mathcal{O}$, how to find the behaviours modelled in $\Gamma$ that are *compatible* with $\mathcal{O}$.
- Diagnosis
  - Set of behaviours
  - Sequences of events that could have occurred on the supervised system
- Diagnosis $=$ Transition system

# Centralised approaches

- *Centralised approaches* $\Rightarrow$ Need of the *Global Model*

$$\|\Gamma\| = \prod_{i \in 1}^{n} \Gamma_i$$

- *Diagnoser approach* [Sampath *et al.*][Rozé *et al.*]
  - Based on an *observer*: a finite-state machine which represents the set of *observable behaviours* from the supervised system
  - Diagnosis information: contained in the states of the observer
  - Advantage: the computation of the diagnosis is efficient (parsing of the observer).
  - Drawback: computation of the diagnoser, good luck!
    - Worst case size of $\|\Gamma\|$: $\geq 2^n$ (MAGDA project $=$ small network $= (n = 57)$)
    - Worst case size of $Diagnoser(\|\Gamma\|)$: $\geq 2^{2^n}$

# Decentralised approaches

- Principle: *Divide and conquer*

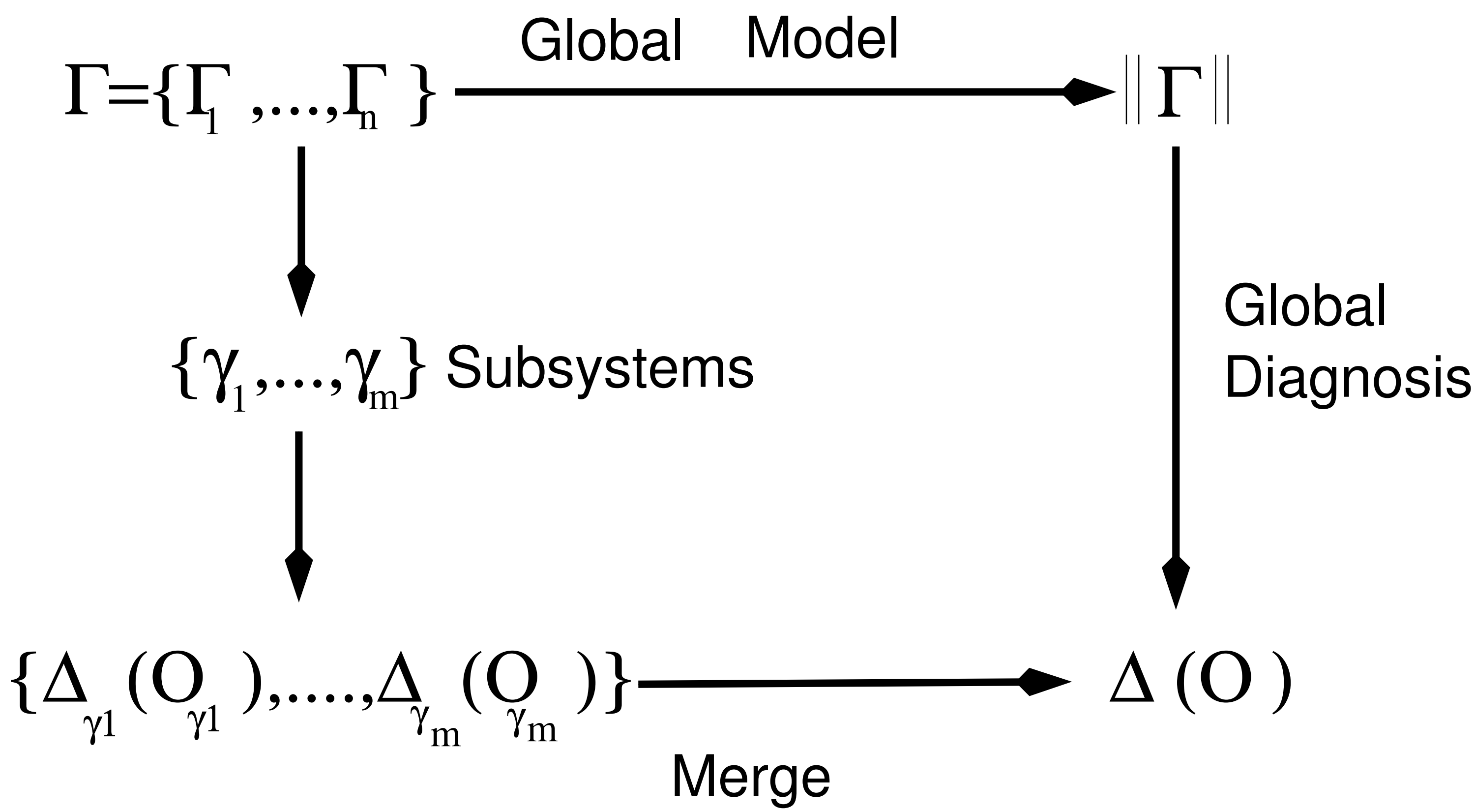
- *Divide*:
  - Computation of a set of *subsystem diagnoses* $\Delta_{\gamma_1}(\mathcal{O}_{\gamma_1}), \ldots, \Delta_{\gamma_m}(\mathcal{O}_{\gamma_m})$
    - Diagnosis which explains *observations* $\mathcal{O}_{\gamma_i}$ *from a subsystem* $\gamma_i = \{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$ by a set of *subsystem behaviours*
    - Explanation based on the hypothesis the subsystem $\gamma_i$ is independent from the others

- *Conquer*:
  - Merge of the *subsystem diagnoses* to get the *global diagnosis*

$$\Delta(\mathcal{O}) = Merge(\Delta_{\gamma_1}(\mathcal{O}_{\gamma_1}), \ldots, \Delta_{\gamma_m}(\mathcal{O}_{\gamma_m}))$$

  - Purpose: diagnosed interactions checking

# Centralised/Decentralised



# Advantages of a decentralised approach

- The *global model* is not necessary
  - Use of *tractable models*

- Supervised systems: distributed systems, well-suited approach
  - More adapted to the evolution, the reconfiguration of a component of the system

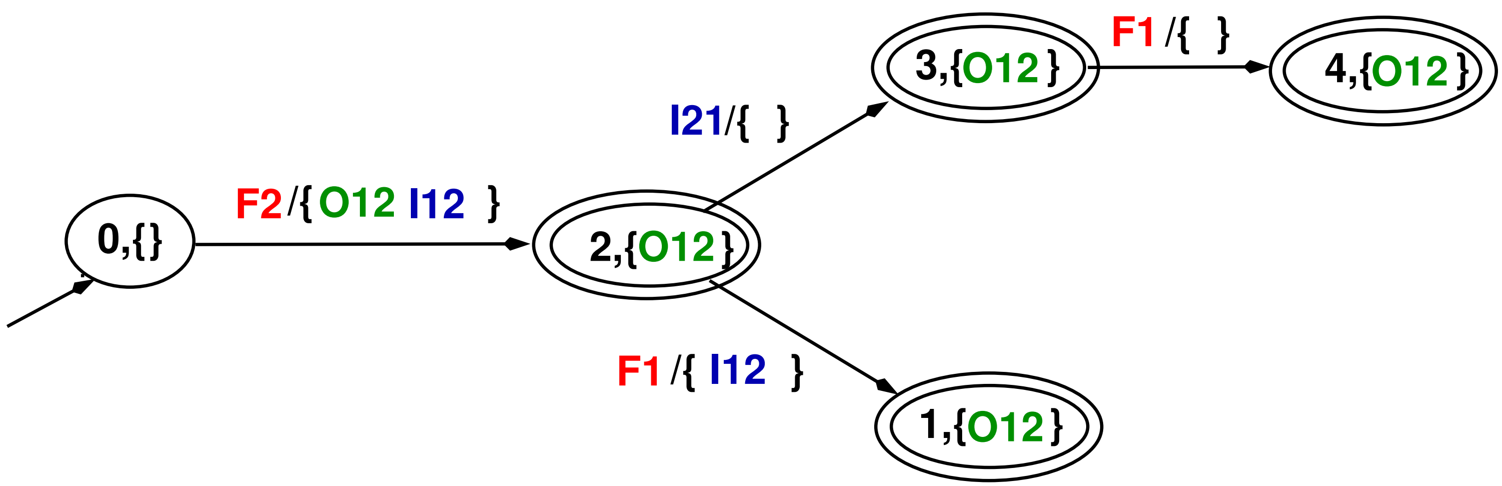
# Decentralised approaches: previous works

- Decentralised and coordinated diagnosers [Debouk *et al.*]
  - One diagnoser by sensor: computes a *local diagnosis*
  - Merge: communication protocols between diagnosers to compute the *global diagnosis*
  - Problem: the local diagnosers still need the computation of $\|\Gamma\|$

- Diagnosis of active systems [Baroni *et al.*]
  - *Simulation* of the decentralised model $\Gamma$ constrained by the received observations
  - *Simulation* by subsystems (*subsystem diagnosis*), and generalisation of the simulation (*Merge*)
  - Disadvantage: *offline method*, can't be used as a monitoring system (offline diagnosis approach)

# Diagnosis representation

- Diagnosis (subsystem and global)
  - Set of behaviours
    - Occurrence of failures and their propagations
- Can be represented by a communicating automaton
- Example: diagnosis of the subsystem $\gamma_1 = \{\Gamma_1\}$ (observation O12)



# Reduced representation

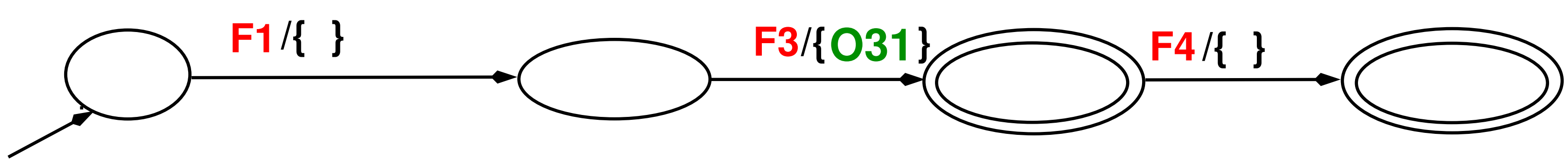
- Diagnosis: set of transition paths
  - the number of paths can be important
    - essentialy due to the concurrency of the system
  - need of a reduced representation

- In the reduced representation,
  - Paths = Event *traces* [Mazurkiewicz 86]
  - equivalent class of *event sequences*
    - equivalence based on *event independency* (concurrency)

- Partial order reduction technique

# Reduced representation: example

- If the diagnosis consists of the following sequences
  1. F1/{} F3/{O31} F4/{}
  2. F1/{} F4/{} F3/{O31}
  3. F3/{O31} F1/{} F4/{}
- If we know that F1/{} and F3/{O31} independent, F3/{O31} and F4/{} independent
- The following path is sufficient to represent the diagnosis
  - by successive permutations of consecutive independent events

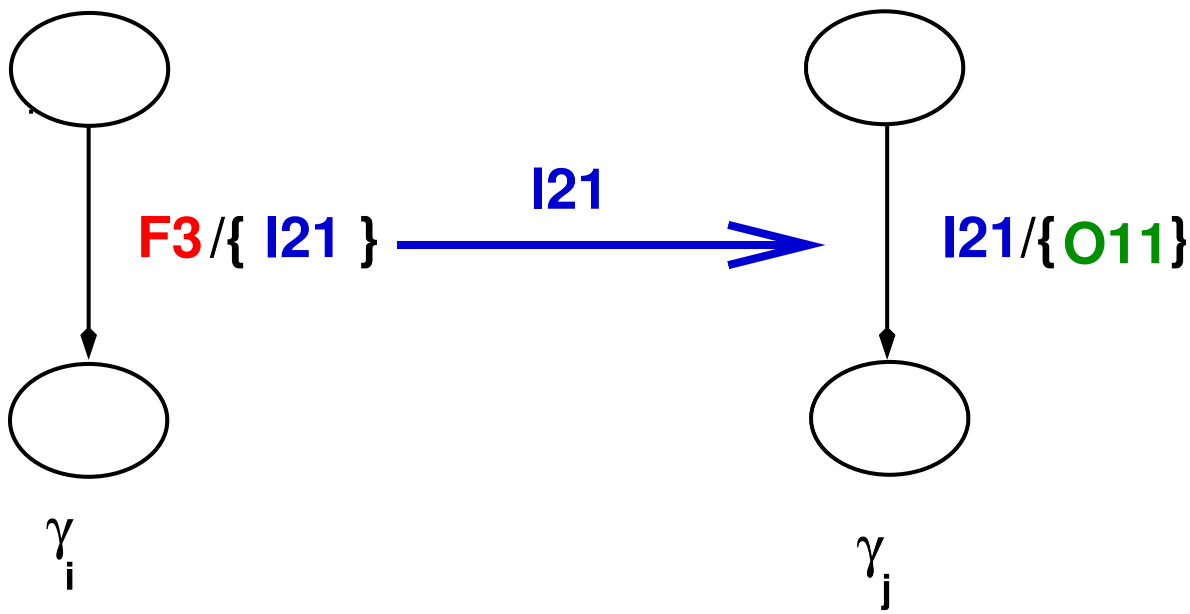


# Local diagnosis computation

- Given a subsystem $\gamma_i \triangleq \{\Gamma_{i_1}, \ldots, \Gamma_{i_k}\}$
- Given $\mathcal{O}_{\gamma_i}$ the set of observations from $\gamma_i$
- Purpose: find the set of paths from $\|\gamma_i\| \triangleq \prod_{j \in \{i_1, \ldots, i_k\}} \Gamma_j$ explaining $\mathcal{O}_{\gamma_i}$
- $\gamma_i$ has been chosen to be tractable
  - A centralised approach can be applied on $\gamma_i$
  - Use of an adaptation of the diagnoser approach [Sampath *et al.*] in order to have an efficient computation
    - Computes paths representing *traces*
    - Noted $\Delta^{red}_{\gamma_i}(\mathcal{O}_{\gamma_i})$.

# Local diagnoses and interactions

- Local diagnosis of $\gamma_i$:
  - Inform about the possible interactions with the *neighbours* of $\gamma_i$
- Interaction: exchange of events, synchronisation



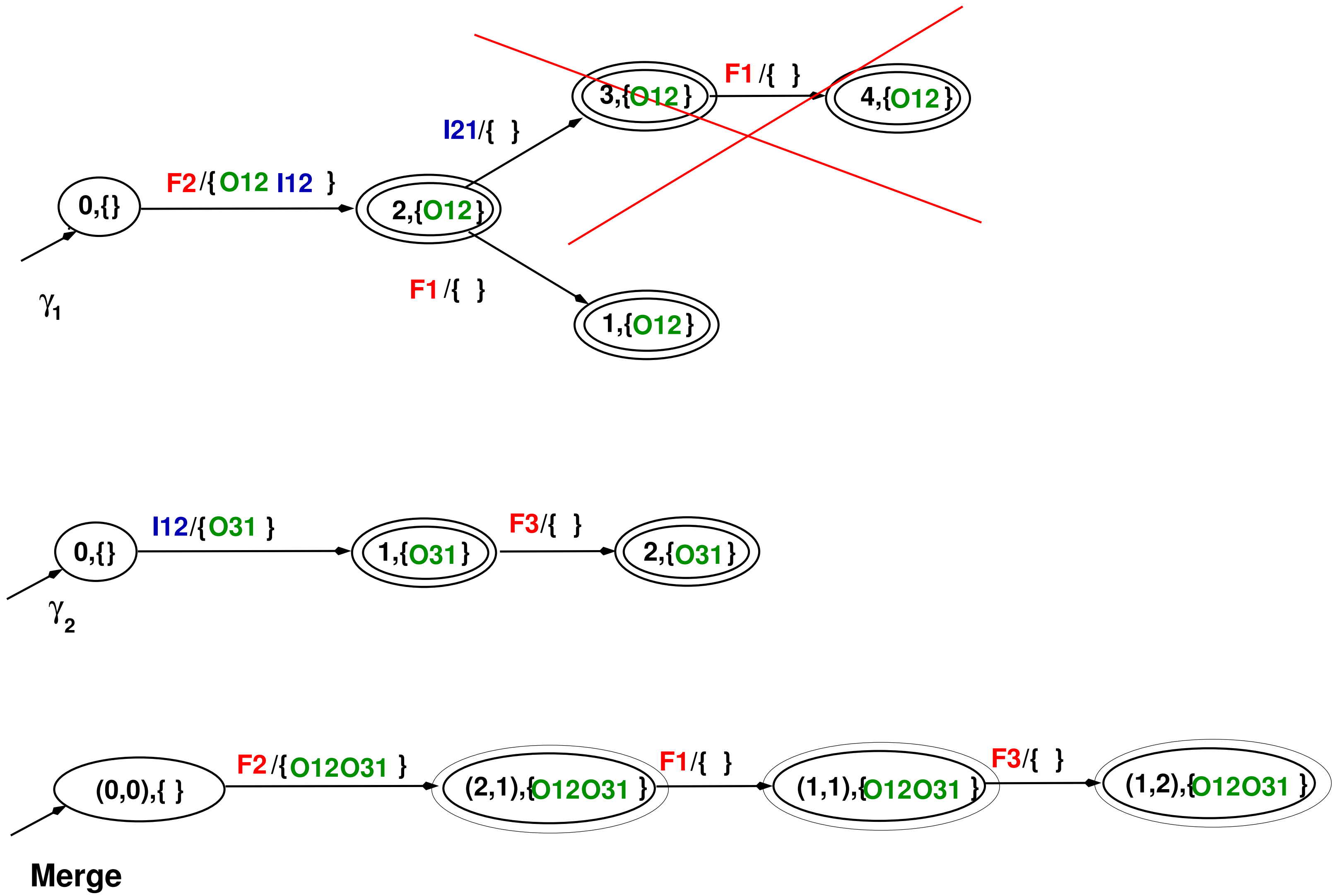
# Merge operation: characteristics

- Purpose: check the interactions between the local diagnoses.
- Merge operation ($\odot$): *synchronised and reduced* product on automata
  - Merge operation recipe:
    1. define a *dependence relation* on the events of the system
    2. mix the results from [Arnold92] (synchronised product of transition system) with the *sleep set algorithm* from [Peled93] (partial order exploration based on the dependence relation) and you have:

$$\Delta^{red}_{\gamma_i \cup \gamma_j}(P_{\gamma_i \cup \gamma_j}(\mathcal{O})) = \Delta^{red}_{\gamma_i}(P_{\gamma_i}(\mathcal{O})) \odot \Delta^{red}_{\gamma_j}(P_{\gamma_j}(\mathcal{O}))$$

where $P_{\gamma_i}(\mathcal{O})$ is the partial order set of observations extracted from $\mathcal{O}$ (projection) which have been emitted by $\gamma_i$

# Merge operation: example



# Merge strategy

- $\odot$ is based on a product operation, it can be not efficient!
  - we have to use it meanly, when necessary.
- We need a plan for the application of the merge
  - Strategy based on the information contained in the local diagnoses
    - What are the diagnoses to merge?
    - *The less I merge, the more efficient I am!*
- The strategy is defined with 2 rules
  - Incompatible path detection
  - Selection of dependent diagnoses

# Rule 1: Incompatible trajectory detection

- Let $E_i$ be the set of exchanged events (interactions) of the subsystem $\gamma_i$ according to its diagnosis

- Every event $e$ exchanged between $\gamma_i$ and $\gamma_j$ is necessary such that:

$$e \in E_i \cap E_j$$

- If not, every path containing $e$ in the diagnosis $\Delta_{\gamma_i}$ is *incompatible*

- **Rule 1**: elimination of incompatible paths before applying the $\odot$ operation on the local diagnoses.

# Rule 2: Selection of diagnoses

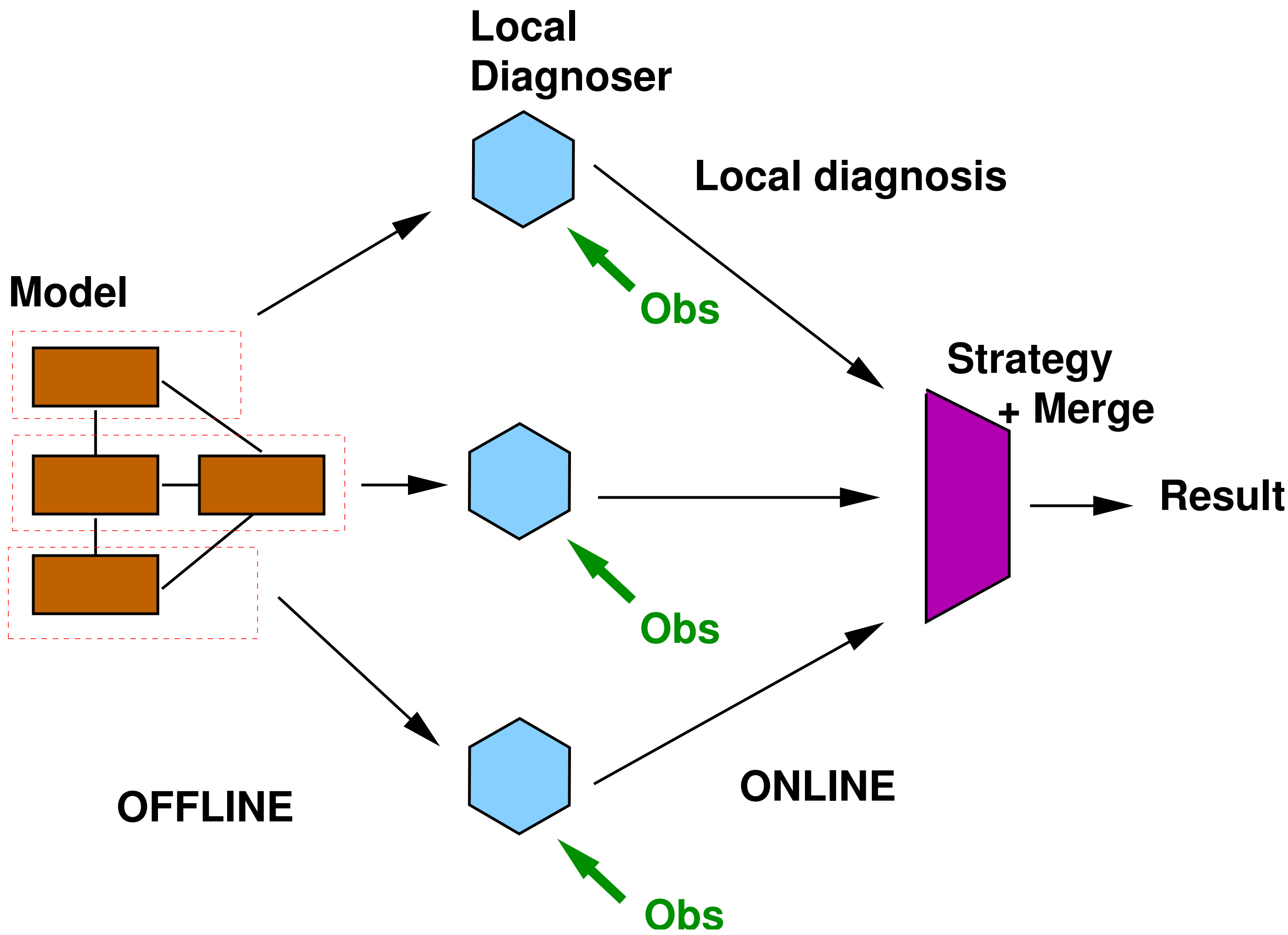
- Basic idea: merging two diagnoses that do not interact each other
  1. is roughly equivalent to make the Cartesian product
  2. is useless, their interactions are not checked

- **Rule 2**: Only merge diagnoses which interact each other

- It is possible to apply the strategy in a parallel way (distributed application)

- The result is a set of *independent diagnoses*.
  1. Each diagnosis gives the explanations of the observations from a part of the system
  2. There is no interaction between the diagnosed parts

# Summary of the approach

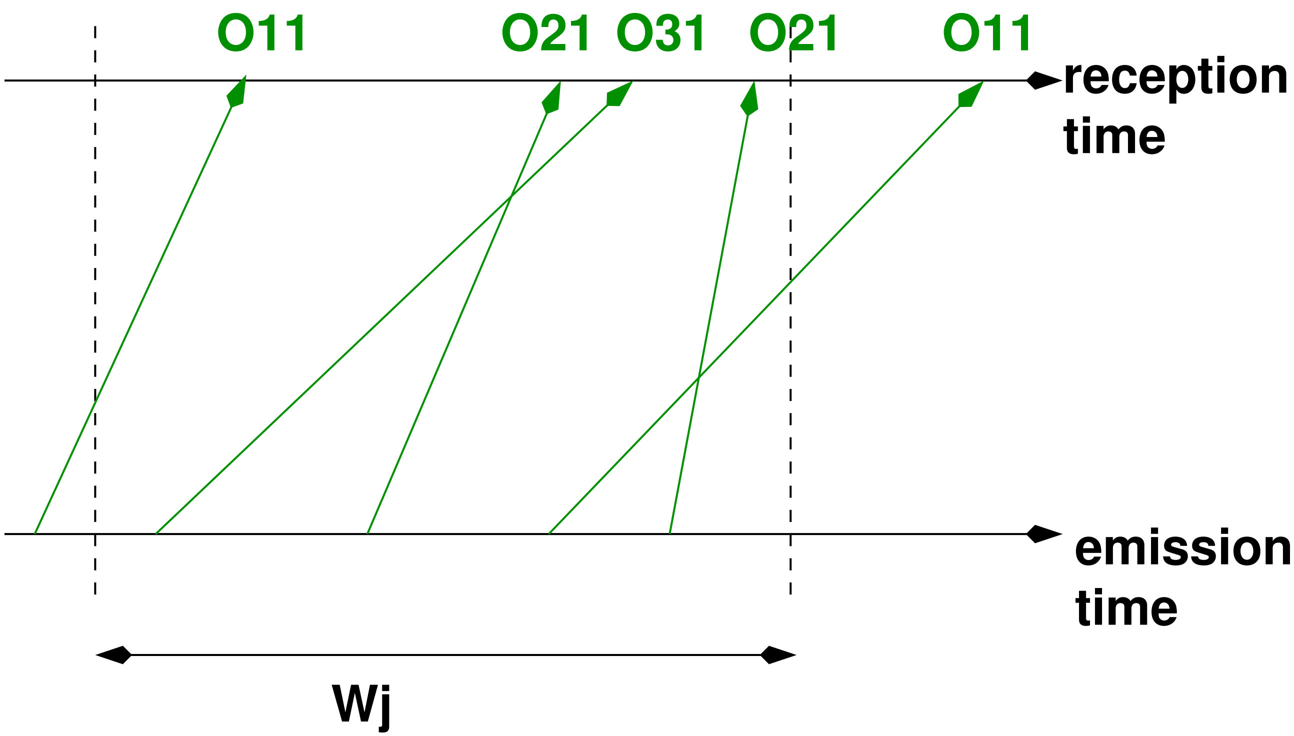


# Incremental diagnosis

- Observations: a continuous flow of alarms
  - Every observation is in a *temporal window* $W_1, \ldots, W_m$
- Given
  - the observations from the window $W_j$
  - the diagnosis explaining the observations from $W_1, \ldots, W_{j-1}$
- How to efficiently compute the diagnosis explaining the observations from $W_1, \ldots, W_j$?
  - Incremental diagnosis computation

# Difficulties

- Generally, at the end of any temporal window, we do not have the guarantee that the received observations can be explained! Some observations might be missing




**O11 is not received during Wj but can be necessary to make a diagnosis**
**The sequence O11 O21 O31 O21 may have no explanation**

# First solution: sound temporal windows

- Detection of *sound* temporal windows
  - An observation is emitted and received in the same window
- Can be detected relying on the observation channel properties
- Incremental diagnosis computation:
  1. From the current diagnosis states at the end of $W_{j-1}$
  2. Computation of the global diagnosis explaining the observations from $W_j$
  3. Refinement algorithm: $\Delta_{1,\ldots,j} \triangleq \Delta_{1,\ldots,j-1} \oplus \Delta_{W_j}$

# General solution

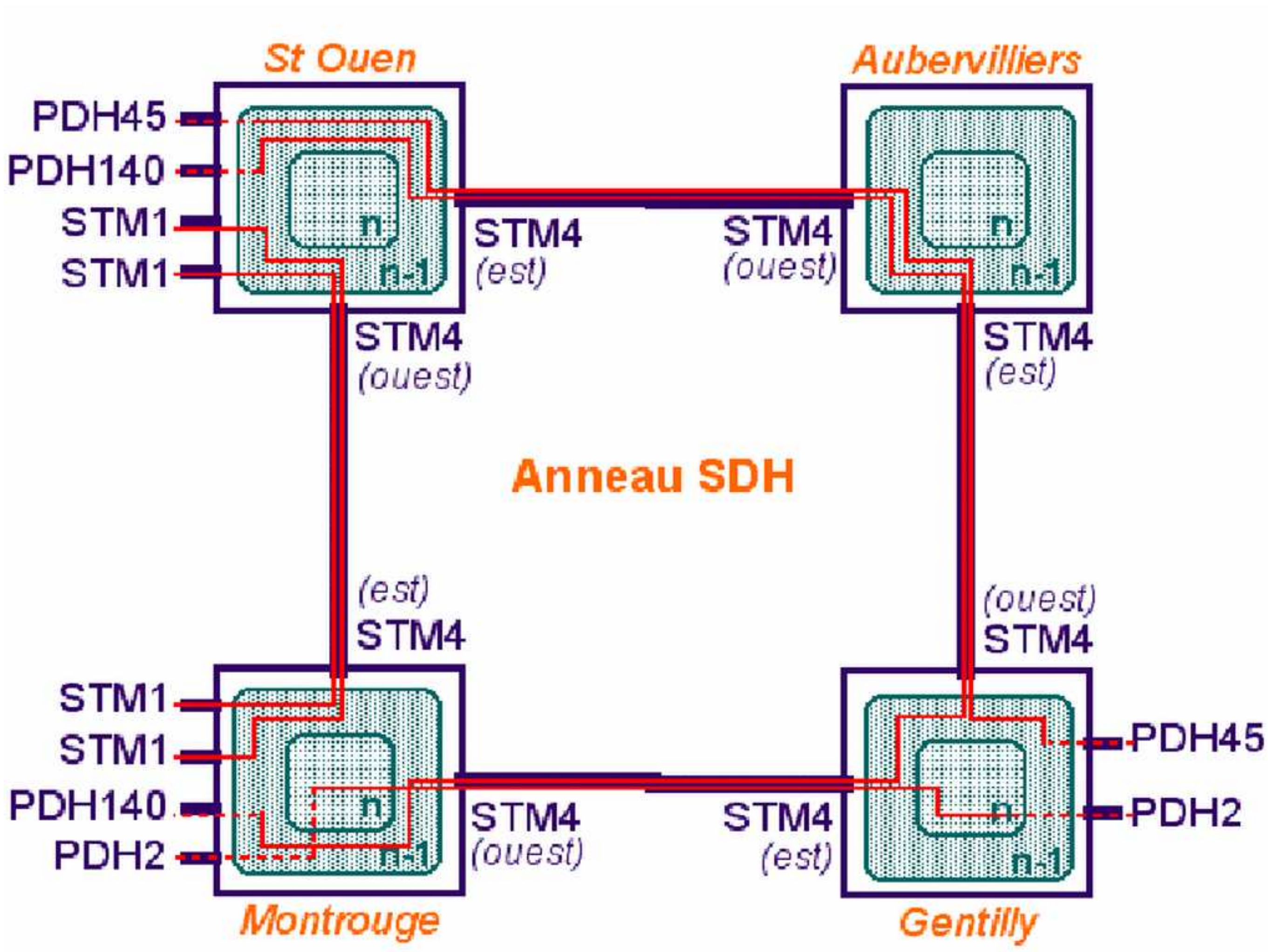
- In the worst case, there is no sound temporal window!
- Need of an *extended diagnosis* $\Delta^{ext}_{W_j}$
  - explains the observations of $W_j$
  - and some *hypothetical* observations, emitted before the end of $W_j$ but not received yet
  - has more explanations than the *real* one
- Incremental diagnosis computation: same algorithm as before
- We have the guarantee that if $W_j$ is sound

$$\Delta^{ext}_{1\ldots j} = \Delta_{1\ldots j}$$

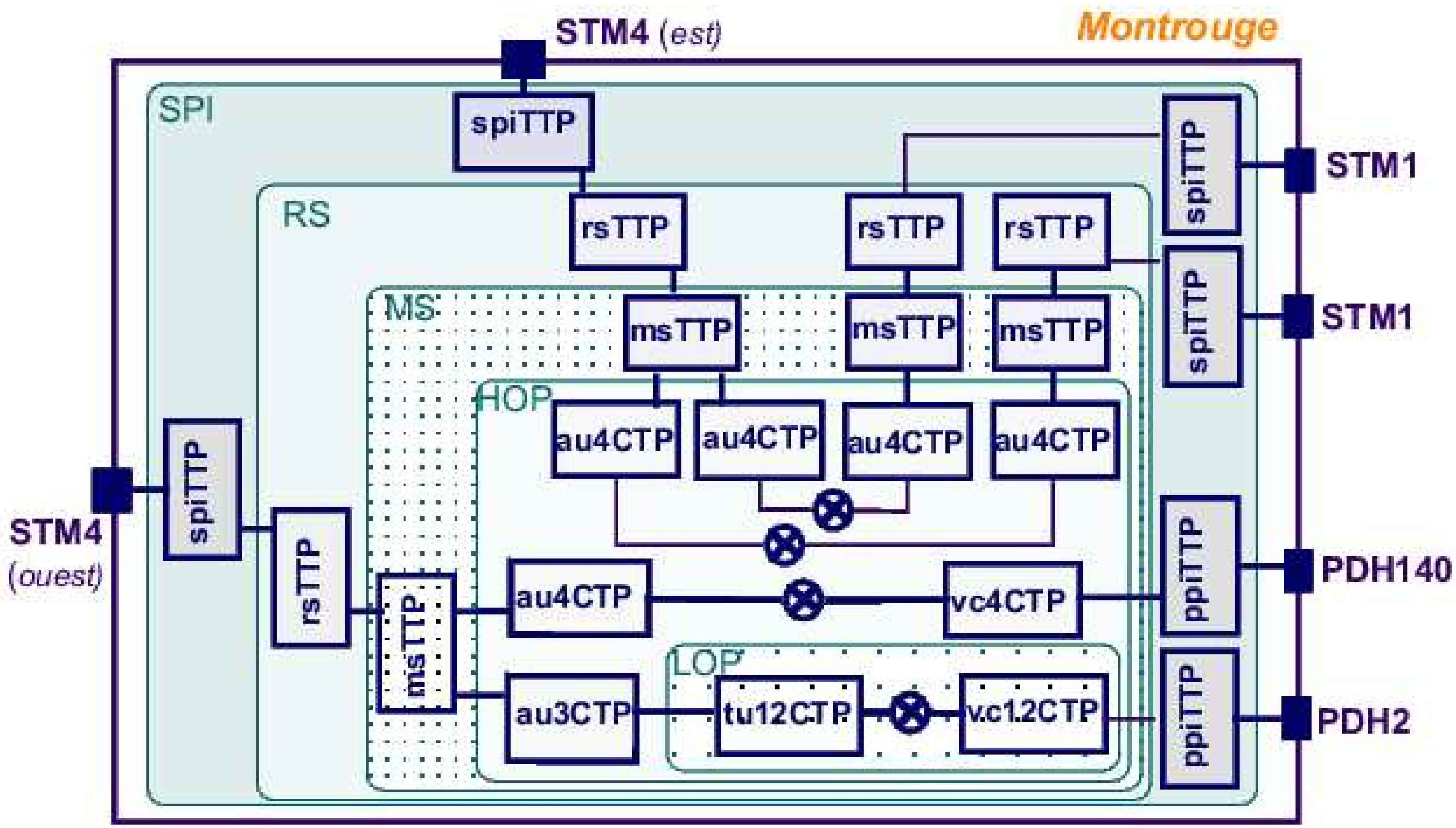
# Transpac network

- French packet switching network
- Experiments done on a sub-part of the network
  - 8 switches, 32 control stations, 2 technical centers
  - Diagnosis difficulty: masking phenomenon
- One studied scenario with 56 alarms
  - Multiple faults diagnosis (masking phenomenon)
  - Result obtained in 8 seconds

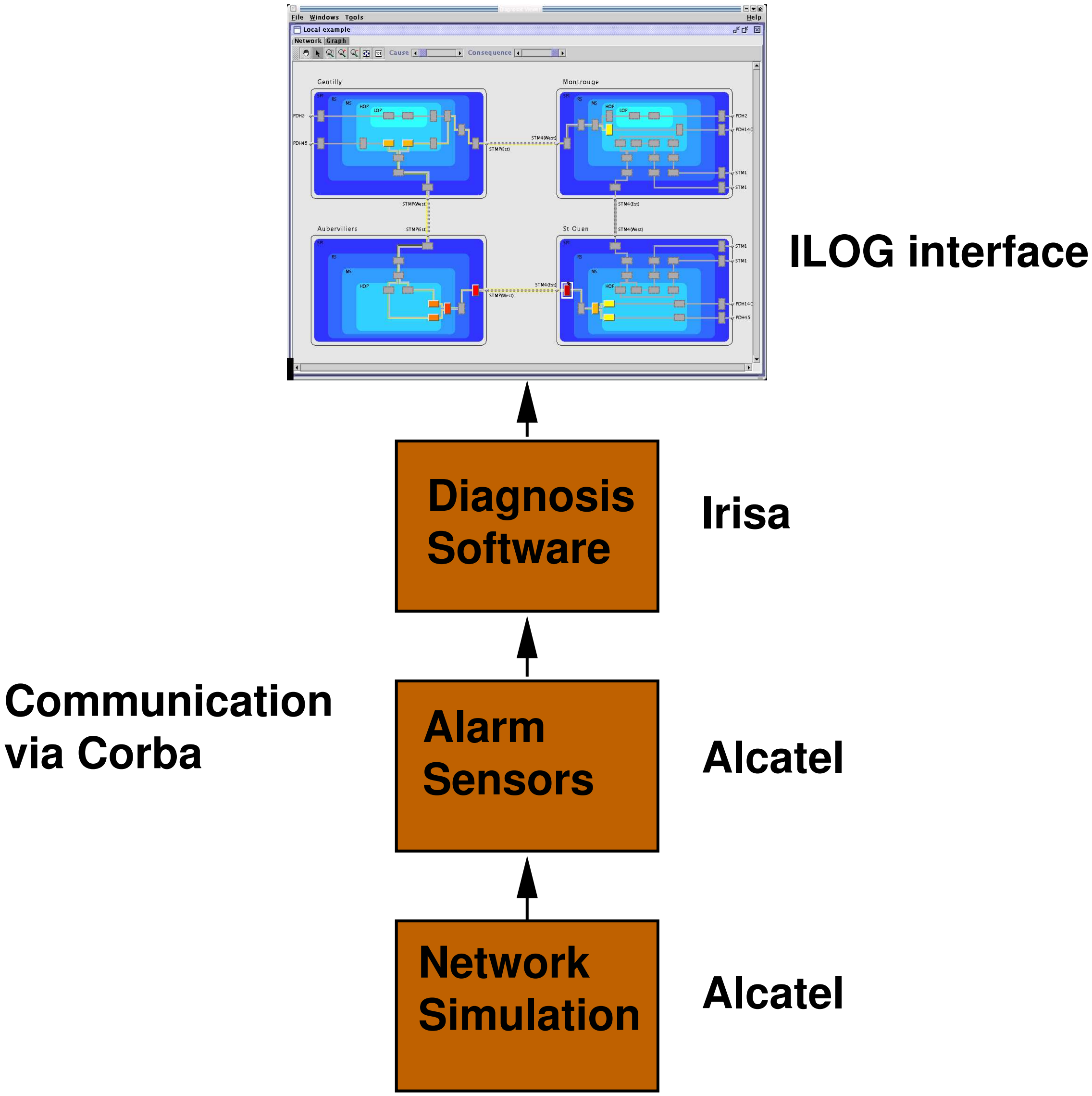
# Magda project: SDH network



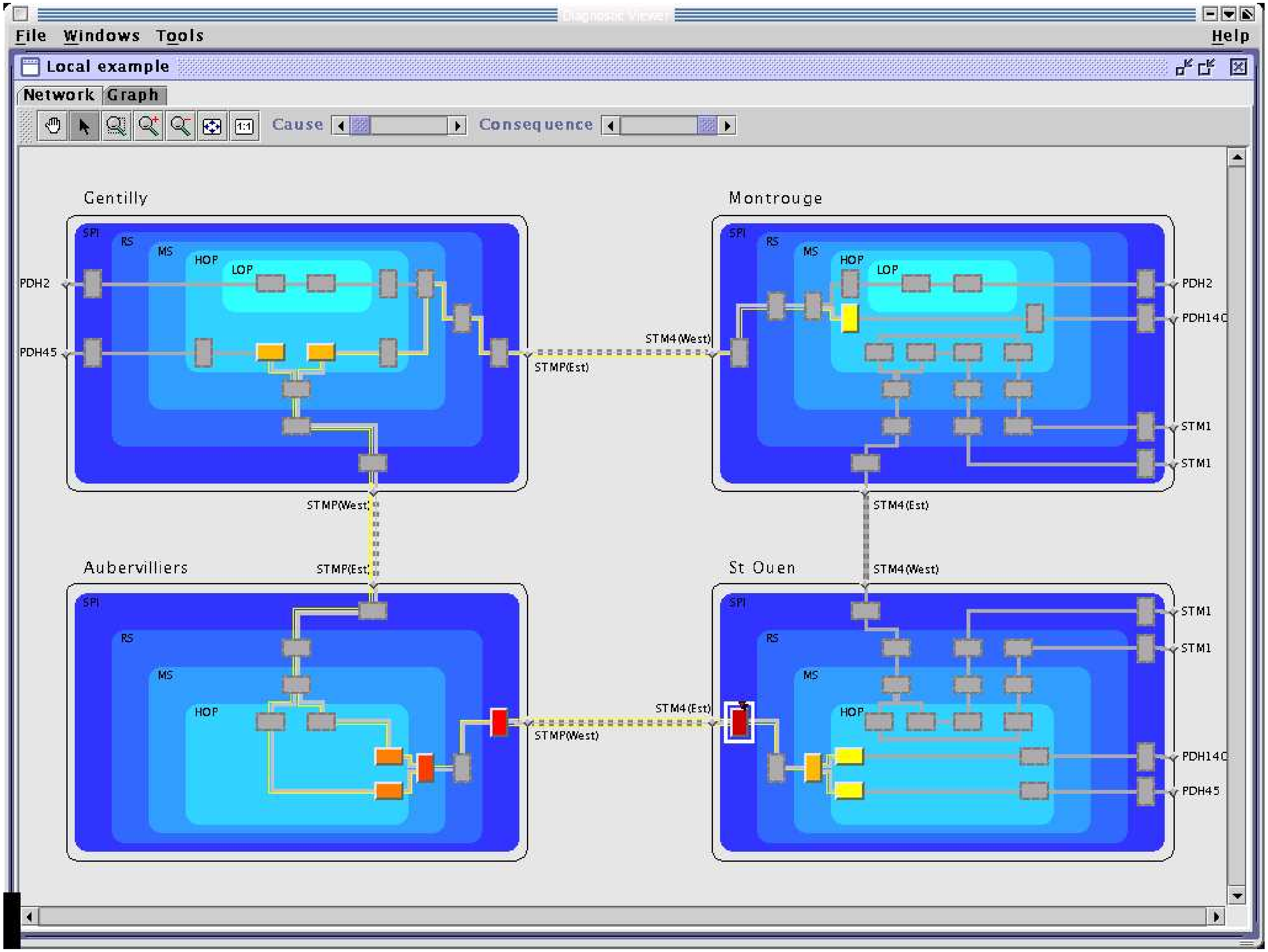
# Magda project: Montrouge ADM



# Magda project: Supervision chain (November 2001)



# Magda project: Interface



# Magda project: studied scenarios

| Scenarios | Strategy 1 | Strategy 2 | Strategy 3 | Strategy 4 |
|---|---|---|---|---|
| 1 | 3s 590ms | 4s 200ms | 16s 540ms | >5mn |
| 2 | 1s 300ms | 1s 300ms | 1mn 52s 770ms | >5mn |
| 3 | 1s 780ms | 1s 910ms | >5mn | >5mn |
| 4 | 1s 600ms | 2s 30ms | 49s 120ms | >5mn |
| 5 | 2s 620ms | 5s 500ms | 5s 430ms | 3mn 45s 600ms |
| 6 | 1s 780ms | 2s 320ms | 24s 240ms | 57s 440ms |
| 7 | 1s 480ms | 1s 700ms | 2mn 54s 920ms | >5mn |
| 8 | 1s 830ms | 3s 90ms | 3s 30ms | >5mn |

- Eight studied scenarios
  - Strategy 1: The previously described strategy
  - Strategy 2: Perturbation of the order of merging
  - Strategy 3: Same as 1 without incompatible path elimination
  - Strategy 4: Same as 2 without incompatible path elimination

# Conclusions

- What a funny challenge it was!!
- Main problem: the use of centralised approches impossible
  - Large scale DES: problem of spatial complexity
- Framework of a decentralised diagnosis approach
  - "Divide and conquer" principle
    - Transfer of a part of spatial complexity to temporal complexity
    - In practice, the number of behaviours explaining a set of observations is very small compared to the number of behaviours of the system
  - "Conquer"
    - Need of merging strategies,
    - Use of diagnosis trace representatives
    - Incremental algorithms

# Perspectives

- How to take benefits from the symbolic representation techniques inside this framework? (BDDs)
- How about the diagnosability test of such systems?
- How about using diagnosability for making diagnosis abstractions?
- How to take into account reconfigurations of the system?
- How to mix with planning approaches (repairing plans)?
  - Large scale autonomous systems