Diagnosing DES in a decentralized/distributed way: some comparison criteria (to be discussed)

Common points :

- the system is a set of interconnected components
- the « components » of the system

are partly in charge of diagnosis computation

Different views of the problem :

- Which system?
- Which kind of diagnosis ?
- Which model?
- Which observations?
- Which techniques for ensuring efficiency of the computation ?

1- Which kind of systems ?

- Discrete-event systems (telecommunication networks, web services …)
  - Reactive systems :

    exogeneous events (commands, faults …) make the system evolve, causing a chain of reactions by propagation, and then the system comes back in a quiescent state
  - Active systems :

    see Marina

- Continuous systems represented as DES (pumps and pipes …)

2 - Which kind of diagnosis :

Who uses the diagnosis ? Who takes the decision?

- Global decision : a supervisor is in charge of monitoring the system and of taking appropriate decisions
  - ⇨ « Decentralized » diagnosis :
    - The observations are often « sent » to the supervisor : delays, loss … i.e need to deal with uncertain observations
    - A global diagnosis must be computed
- Local decision : no supervisor, the decision are taken by the components (less coordinated decisions)
  - ⇨ « Distributed » diagnosis :
    - Local observations : less risk for uncertainty
    - Local diagnoses (no need for computing a global one) – refined by communicating with other components

2. Which kind of diagnosis : Off-line versus on-line ?

- Off-line : all the observations are known when starting the diagnosis computation

- On-line : the observations are collected in parallel with the diagnosis computation
  - It is always the case in a distributed context ?
  - It is mainly a question of « completeness of the observations » ? : you know / do not know whether some observations are going to arrive later …
  - It is also a question of efficiency (real-time)?

2 - Which kind of diagnosis : what are we looking for?

- Diagnosis as
  - Localization : finding the guilty components
    - Diagnosis as sets of components
  - Identification : finding the faults
    - Diagnosis as sets of possible faults
  - Tracking : finding the trajectories
    - Diagnosis as sequences of events
    - Diagnosis as automata

3 - Which kind of model ? Global versus local

- In general, a component-oriented model
- Two ways :
  - The components share a global behavioral model of the system: a global model
    - The observations are distributed but not the model
    - The components exchange local diagnoses via communication protocols
    - Important problem : intractable size of the global model
  - The components are only aware of their own behavioral model: local models + links between the components
    - Need for synchronization

3 - Which kind of model ? Formalism

- Formalisms adapted to DES :
  - Automata, Process algebra
    - Useful composition operation
    - Problem with concurrency
  - Petri nets
    - Well-adapted to concurrent processes
    - Share of ressources
    - Less component-oriented ??
- Rep. of asynchronous/synchronous communications
- Plus
  - Representation of preferences (probabilities)
  - Representation of temporal information

4 - Which kind of observations ?

- Dealing with which uncertainties :
  - On emission dates (no synchronized clocks) :
    - Only partially ordered observations
  - On safety of communications :
    - Possible loss of observations
    - Possible change in the content of observation
  - On the sensors :
    - Uncertainties on the observed values

- Adapted formalisms for representing observations
  - Partial ordered sets
  - Automaton

5 – Improving the efficiency of diagnosis computing ?

- Compiling the model:
  - Diagnoser
  - Specific data structures
  - But is it so important when local diagnoses, i.e in general computed for « small size » components ?
- Using model-checking techniques
  - BDD
  - Partial order semantic to efficiently represent concurrency in automata formalism
- Adequate strategy for « synchronizing » the local diagnoses (merging strategy)

Examples

- Decentralized (supervisor) + global model :

  Lafortune, Debouk, Sengupta et al.

  – Three protocols according to shared information

- Decentralized (supervisor) + local model :

  Pencolé et al. , Zanella-Lamperti

  – Model formalism : automata (communicating automata)
  – Diagnoser + model-checking techniques / specific data structures
  – Merging strategy based on interactions between components
  – Incremental algorithm (temporal window)

- Distributed (no supervisor) + local model :

  Fabre et al. / Boel et al.

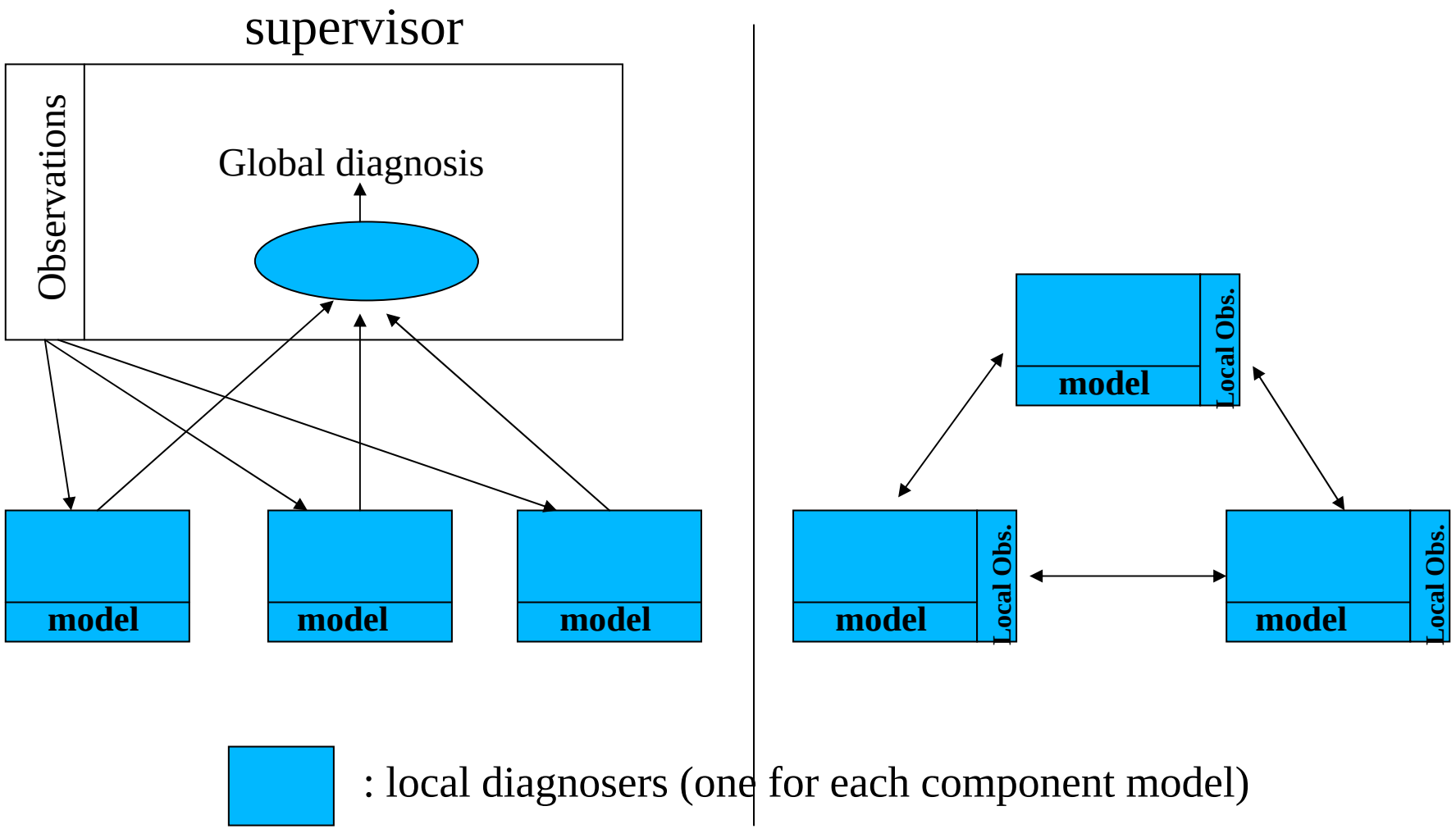
  – Model : Petri nets or rules (pièces) + probabilities in the models
  – Viterbi-based algorithm / abductive algorithm

Other features

- Dealing with change of components models, of components connections (reconfigurable systems)
- ?

Two architectures

Decentralized | Distributed



« décentralisées »

4– modèles locaux et observations?

- Approches avec superviseur :
  - Différences entre observations émises (x capteurs locaux) et les observations reçues (1 capteur global - séquence)
  - Remarque : On connaît les dates de réception des observations reçues par le superviseur, quelquefois les dates d'émission mais en absence d'horloges synchronisées, cela ne fournit qu'un ordre partiel capteur / capteur (composant/composant)