Distributed detection based on chronicle recognition

Amine Boufaied,Audine SUBIAS,Michel Combacau

subias@laas.fr

IMALAIA – Paris
2005/10/14

Our context

- **Distributed systems:**
  - distributed process
  - distributed control and monitoring architecture
  - set of monitoring sites (*diagnosers*) with observers

  (M. Zanella): No consideration of overlapping between observers

- **Process failure detection**
- **On-line approach**
- **Model based approach**
- **Time model : chronicles**

The focus : communication aspects between diagnosers

- **No assumptions on the message ordering**
- **No virtual global clock**
- **No loss of events**

- **Problem of uncertainty due to the communication delays**
  - **clocks synchonisation :** stamping and reordering the messages
  - **specific algorithms for reordering of messages:**
    - need of a coordinator
    - no consideration of the duration between events occurrences

The objectives: an approach for….

- **The verification of timing and sequencing relationships between events**

  To use for failure detection purposes

  To improve classical techniques of clocks synchronisation and messages ordering

- **Addressing the problem of the cost related to the uncertainty**

  by checking the membership of inter-events duration to specific intervals

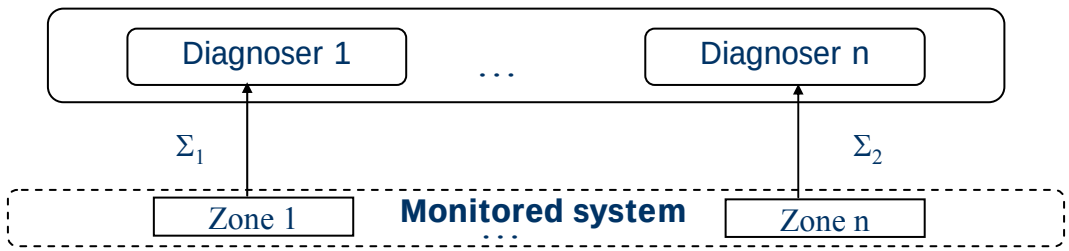
Time representation: the assumptions

- **A time stamp can be assigned to each event**
- **Only partial ordering of events is known**
- **An event is characterized by its occurrence date and has no duration**
- **Timing information encoded in terms of duration constraints**
- **Duration :** difference between the occurrence date of two events

Distributed detection and chronicle

- **Distributed process and distributed monitoring system**

- **The detection function** monitors the process evolution through chronicle recognition



- **Chronicle: specific sequences of events related by timing constraints**

- **Failure symptom detection:**

  timing constraints of normal evolution**: constraint violation**
  timing constraints of erroneous evolution**: constraint verification**

Preliminary notions

- **Event :** associated to the begining/the end of an activity executed by the controlled process

- **Occurrence date:** date of an event

  ✓ **local clock** associated to each diagnoser

- An event is dated in one and only one **local time referential** linked to one monitoring site

by The **occurrence function** noted **O** is defined by :

$$O : \Sigma \rightarrow Q^{+}$$

$$e_i \rightarrow O(e_i)$$

Preliminary notions

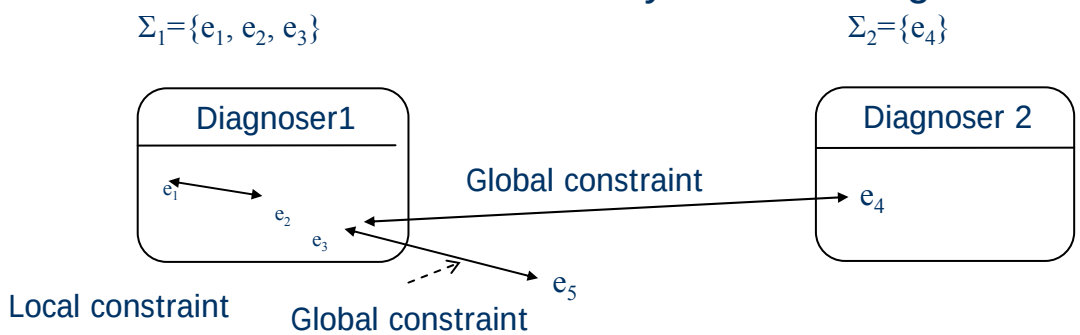
- **Constraints between events:** relationships expressed by a duration between events (causal relation of the application e.g. transport duration)
- Binary constraint:
  - **precedence type constraint** $O(e_i) < O(e_j)$
  - **interval type constraint** $d_{j,i} \leq O(e_j) - O(e_i) \leq f_{j,i}$ noted C_{ji}

    avec $d_{j,i}$ et $f_{j,i} \in Q^+$
- n-ary relation : **window admissibility constraint**

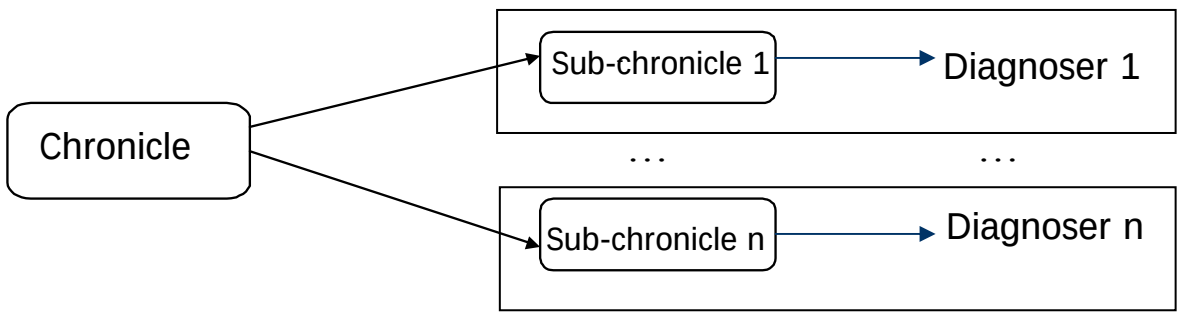
  $$d_i \leq \min_j(O(e_j)-O(e_i)) \leq f_i \text{ , noted } D_i, \text{ with } d_i, f_i \in Q^+$$
- **Chronicle:** a set of events E and a set of timing constraints between these events

Preliminary notions

- **local constraints:** link events dated by a same diagnoser i
- **global constraints:** link events dated by different diagnosers



- notion of **sub-chronicle :** set of events, local constraints, global constraints



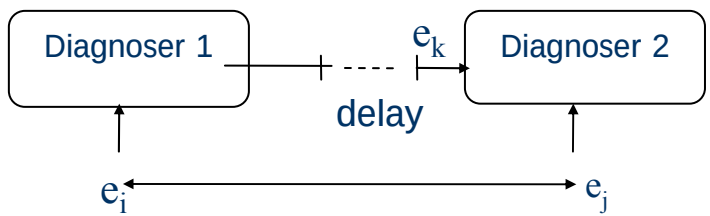
The distributed detection function

- **chronicle recognition:** checking that all the constraints are satisfied

  ⇨ **recognition of the n sub-chronicles**

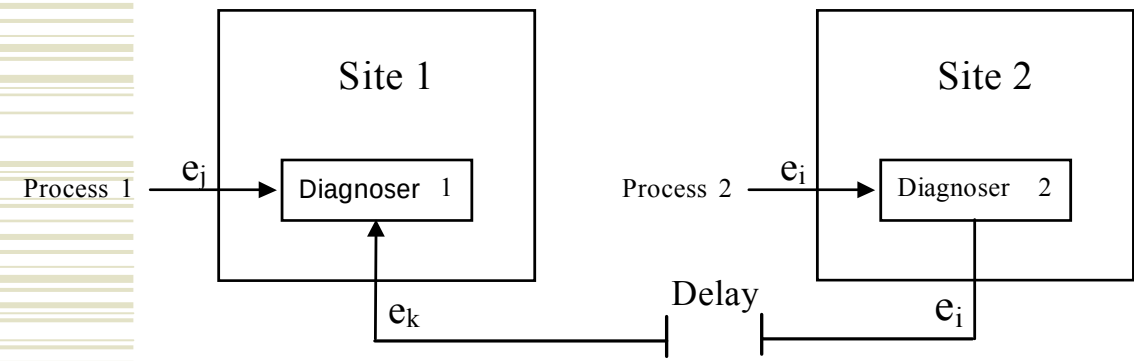
- **sub-chronicle:** set of events, local constraints, global constraint

⇨ **How to verify a global constraint with communication delay consideration?**



- **Hypothesis**: the possible values of the delay are uniformly distributed on the interval $[\delta_m , \delta_M]$

Global constraint verification



Problem:

$d_{j,i} \leq O(e_j) - O(e_i) \leq f_{j,i}$

Delay $\Delta = [\delta_m \; \delta_M]$

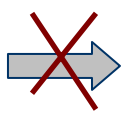
$O(e_j)$, $O(e_k)$

O(ei): OK ?

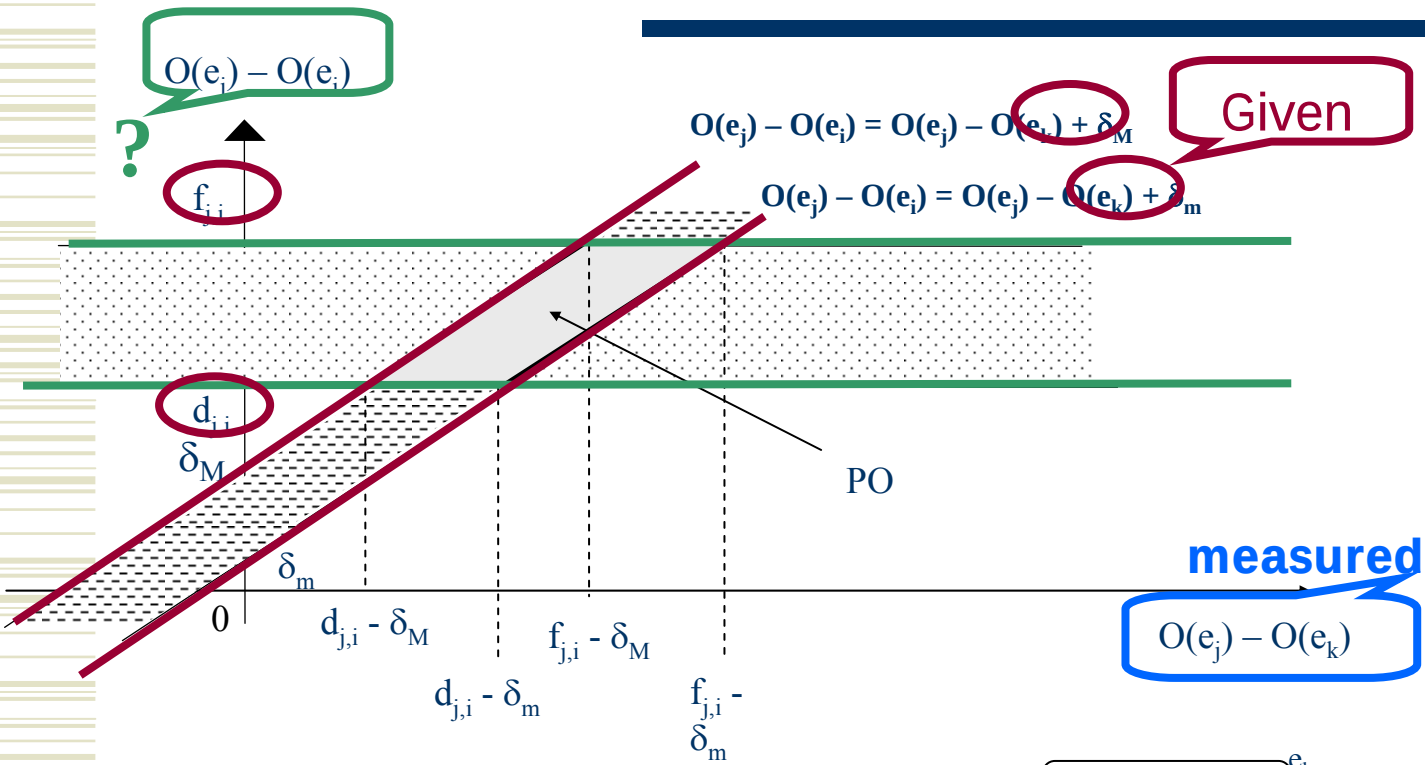
- A new formulation of the initial global constraint

$$O(e_j) - O(e_i) = (O(e_j) - O(e_k)) + \Delta = (O(e_j) - O(e_k)) + (O(e_k) - O(e_i)),$$

$$\mathbf{O(e_j) - O(e_k) + \delta_m \leq O(e_j) - O(e_i) \leq O(e_j) - O(e_k) + \delta_M} \text{ (local constraint)}$$

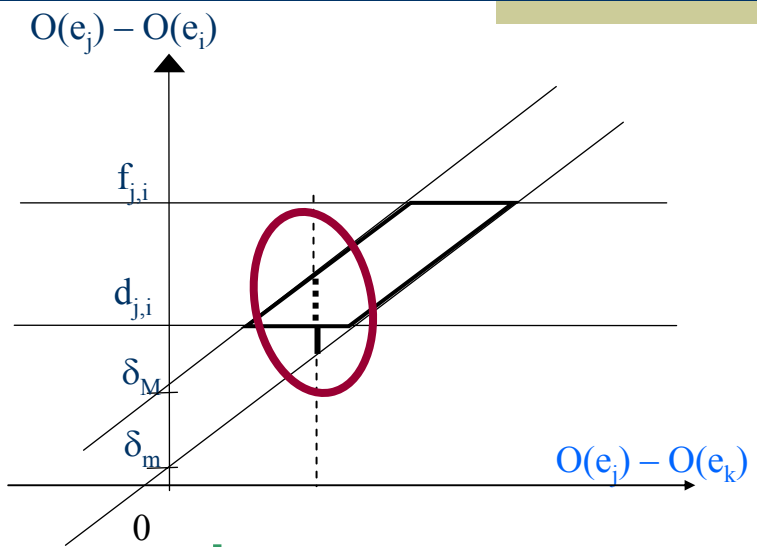
- Verification of a local constraint  verification of the global constraint

Global constraint verification

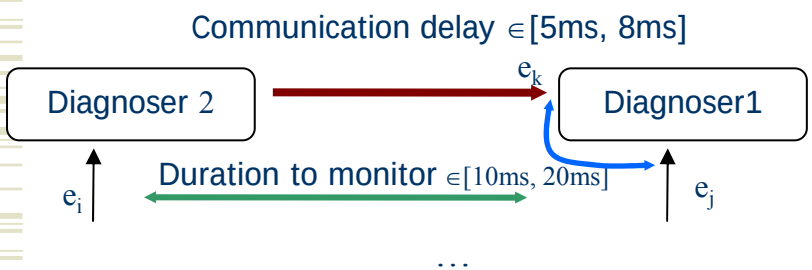


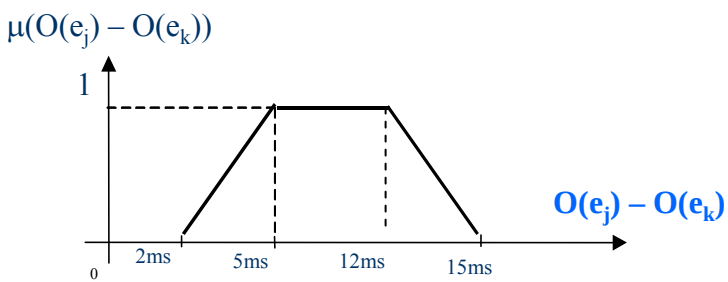
Global constraint verification

- **Quantify** among the set of possible durations $O(e_j)$-$O(e_i)$ / verify the specified constraint



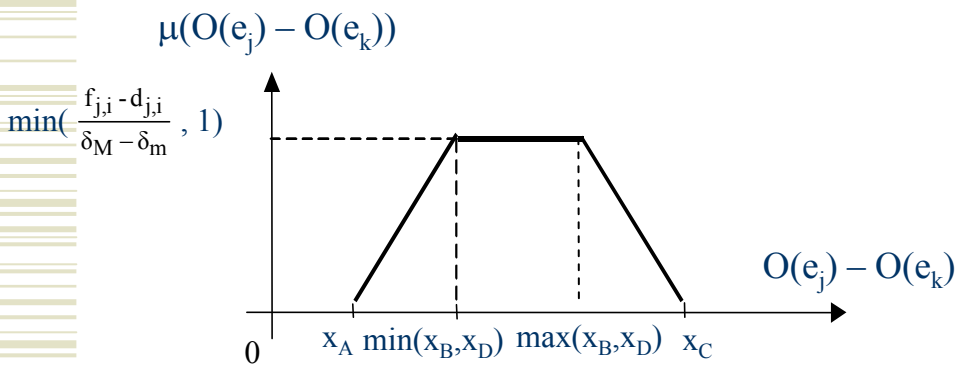
Possibility function to check a constraint





Global constraint verification

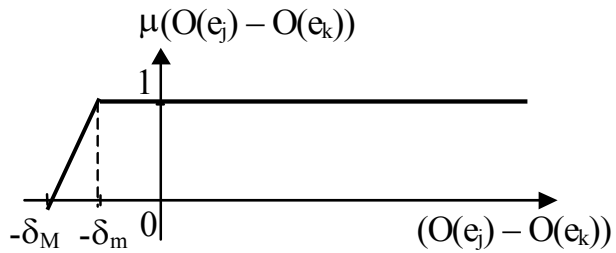
Interval type constraint



$X_A = d_{j,i} - \delta_M$, $X_B = d_{j,i} - \delta_m$,

$X_C = f_{j,i} - \delta_m$, $X_D = f_{j,i} - \delta_M$,

Precedence type constraint



Window admissibility type constraint :

- Analogous result
- Conjunction of several interval constraints
- Problem of global order of events occurrences dated on different diagnosers to determine the last event

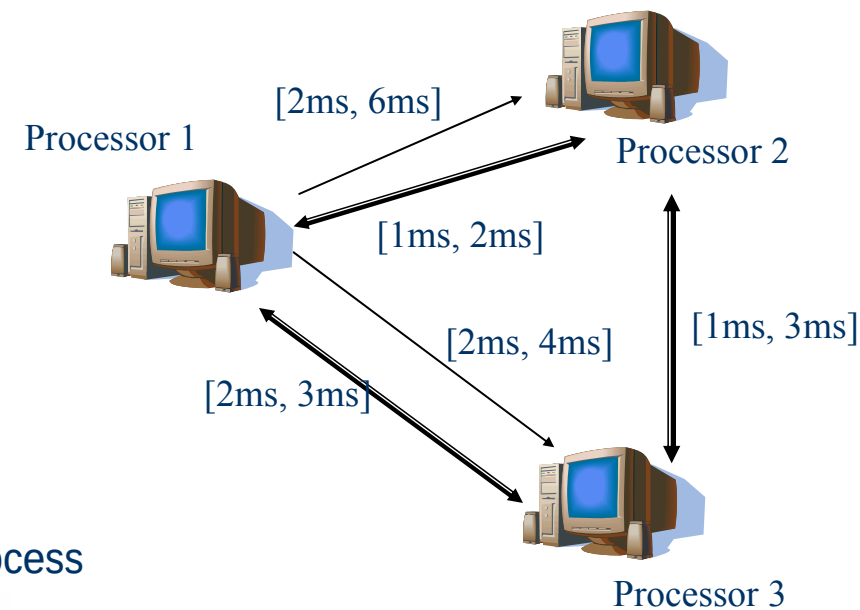
Tasks scheduling on multiprocessors architecture

- No tasks duplication
- 3 processors
- 3 tasks
- End of task i : event e_i

- The constraints:
  - Interval type
  - Global
  - Normal evolution of the process

$$C_{21} : d_{2,1} < O(e_2) - O(e_1) < f_{2,1}$$

$$C_{31} : d_{3,1} < O(e_3) - O(e_1) < f_{3,1}$$

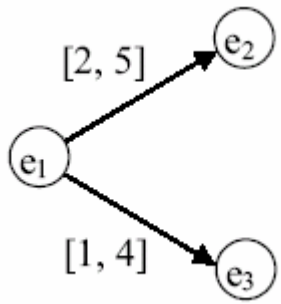


Constraint bounds

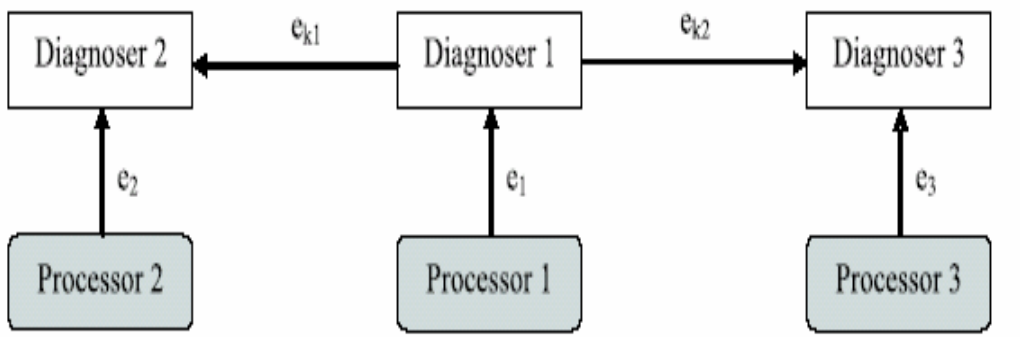
Communication delays

Tasks scheduling on multiprocessors architecture

- The chronicle (global model)
- The 2 sub-chronicles (local models)



- Monitoring architecture and message exchanges



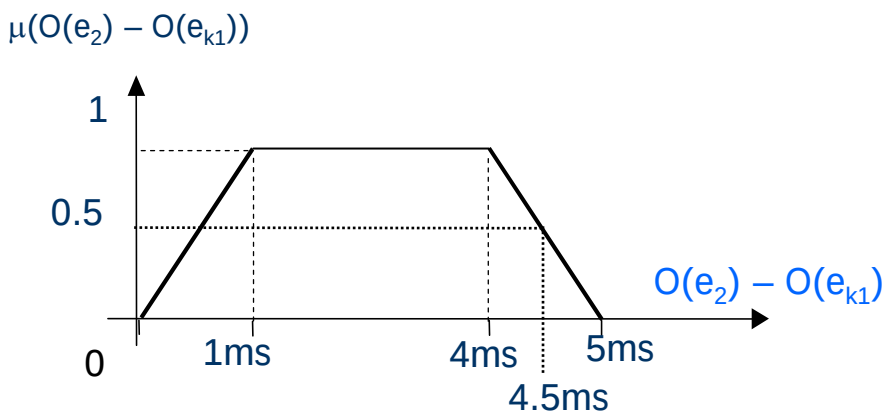
Tasks scheduling on multiprocessors architecture

⇨Monitoring of the duration between the end of task1 and the end of task2

C_{21}: $d_{2,1} < O(e_2)-O(e_1) < f_{2,1}$ **with $d_{2,1}=2ms$ and $f_{2,1}=6ms$**

$\Delta=[1ms,2ms]$

Measure: $O(e_2) - O(e_{k1})$



$O(e_2)-O(e_{k1})=2.5ms$ ⟶ No failure symptom detection

$O(e_2)-O(e_{k1})=5.5ms$ ⟶ Failure symptom detection + localisation

$O(e_2)-O(e_{k1})=4.5ms$ ⟶ **Possibility value: no certitude**

On the fuzzy notion….

- **How obtain a certitude on a global constraint verification?**
  - Fix a verification threshold : how? Use of historical data
  - Ask an operator to check the process state (depends on the process type…)
  - Introduce a **cooperation between diagnosers**

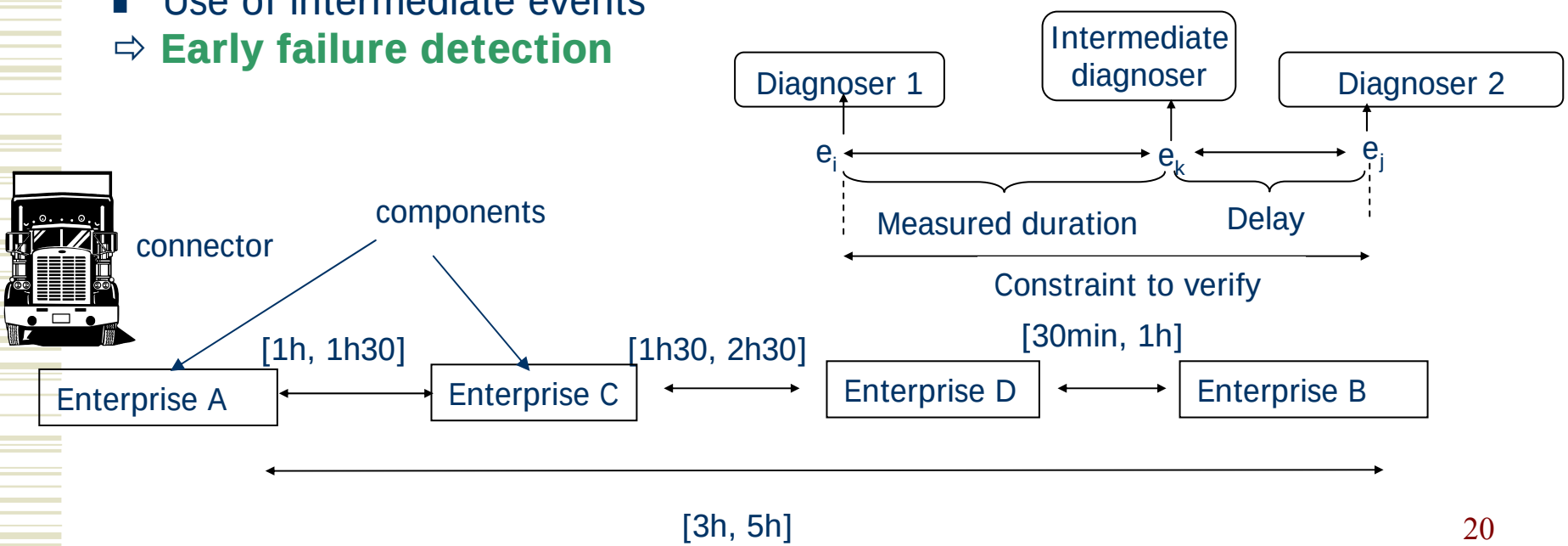
Conclusion

- A monitoring architecture based on autonomous and cooperative sites: diagnosers
- Time model of the process evolutions: chronicle and sub-chronicles
- Communication delays consideration
- Possibility functions to verify a constraint
- Detection function: fuzzy verification of a timing constraint
- Suitable for systems / operational constraints are comparable to delays

Conclusion

- Application to computer network systems
- Application to transportation systems or supply chains
  - Transportation delays between sites vs communication delays
  - Use of intermediate events

⇨ **Early failure detection**



Future Works

- To develop an approach to quantify the performance of the detection function according to the delay
- To consider delays with uncertain bounds or non uniformly distributed
- To integrate the detection function in a whole monitoring system and to consider others supervision and monitoring functions: diagnosis (localisation and explanations), recovery

Distributed detection based on chronicle recognition

Amine Boufaied,Audine SUBIAS,Michel Combacau

subias@laas.fr

IMALAIA – Paris
2005/10/14