7. Supervision and monitoring

- Monitoring the plan execution and reactive scheduling
- Monitoring the behaviour of the physical devices

"Batch processes and hybrid systems" aspect

---

Supervision and monitoring (1)

What is its purpose?

- **Implementing the manufacturing plan**
  - send the right commands at the exact moment as indicated by the plan
  - detect any deviation and implement reactivity

- **Monitoring the physical system**
  - update the current state of the system after each operation
  - detect any abnormal behaviour of the physical system and reactivity to failures
  - guarantee the security of humans, environment, machines

- **Supervises the preventive maintenance**
  - when machines are stopped for maintenance
Supervision and monitoring (2)

Architecture: real-time interface: management / control

Previsionnal plan

Management levels

Controlling the flow of products

Current state

Controlling the resources

Local control levels

Command

Supervision and monitoring (3)

Controlling the flow of products:

- **Requirement analysis**
  - what are the need of reactivity and flexibility?
  - what are the required pieces of information (indicators) for real time decision?

- **Design solutions for detecting deviations and decision aid**
  - how interconnecting previsional scheduling and real-time decision

- **Performance evaluation, validation of the solution**
  - tools for simulation, analysis of social organisations
Supervision and monitoring (4)

Controlling the resources:

- **Requirement analysis**
  - what are the faults, the errors, the failures, their causes, their effects?

- **Design solutions for detecting failures and decision aid**
  - detection, diagnosis, decision aid for recovery and compensation
  - models: structural, functional, behavioural

- **Security constraints, protection of humans & environment**
  - prevent the sequences of unexpected events leading to disasters
  - supervisory control (Ramage and Wonham)

Supervision and monitoring (5)

General control system architecture:

![Control System Architecture Diagram]

*ICATPN, Advanced Tutorial "Petri nets and production systems", Lisboa, 23 June 1998*
Model for Supervis. and monitor. (1)

Utilisation of the model (Petri net based model):

- **Representation of the current state of the manufacturing sys.**
  - machines and parts denoted by tokens, FMS state = current marking

- **Elaboration of list of possible decisions**
  - list of enabled transitions, tuples of tokens enabling a transition (coloured PN)
  - transitions in conflict, transitions which can be concurrently fired

- **Detection of any abnormal behaviour**
  - a system state change does not correspond to an enabled transition - no update
  - a sequence of the plan is inconsistent: corresponding trans. not enabled

Model for Supervis. and monitor. (2)

Utilisation of the model (Petri net based model):

- **Action = transition firing / only enabled + event or decision**
  - The token player for real-time decision and detection

- **Controlling the flow of products**
  - enabled transitions = what it is possible to do in the current state
  - plan = the desired decisions

- **Controlling the resources**
  - enabled transitions = normal next events (with respect to model of correct behaviour)
  - event = actual state change of the system
Model for Supervis. and monitor. (3)

The two roles of the token player:

- Constraints 1 plan (desired)
- Constraints 2 current state of phys. sys.
- Event (message from LAN)
- Fault
- Constraints current marking of Petri net model
- Real-time decision (control)
- Detection and update mechanism

From discrete event to hybrid (1)

Utilisation of the model (Petri net based model):

- The model is a pure discrete event model
  - The model is executed step by step, controlled by decisions and events
  - Events are uniquely determined by discrete states, no explicit time
  - Detect any inconsistency within message sequences (precedence relations) in the control system = a distributed control architecture

- Decision
- Constant state event
- Variable state event
- Time event
- Fill the reactor
- Heat until temp = param.
- Stir during 5 minutes
**From discrete event to hybrid (2)**

Utilisation of the model (Petri net based model):
- \( t_1 \): receive decision (management), check if enabled, send command "fill the reactor"
- \( t_2 \): receive event (from sensor or local cont.), check, send command "heat" (if plan ok)
- \( t_3 \): receive event (from sensor or loc. cont.), check if enabled, send "stir" (if plan ok)
- \( t_4 \): receive event (from real-time clock), check, send "stop"

![Diagram](Image)

**From discrete event to hybrid (3)**

Expliciting time:

- **To each activity (place) a min and a max duration is attached**
  - determined from the behaviour of the physical system

- **To each transition a time window is attached**
  - from the management level, earliest starting time, latest starting time

- **The model can be simulated**
  - mean value, stochastic, earliest time or shortest duration, latest time or longest dur.
From discrete event to hybrid (4)

Detection with time:

- **Detect inconsistency within sequence, and out of time window**
  - after trans. firing, compute current time interval, intersection with plan interval
  - at the occurrence of the event, check that it is in the time window
  - constraint propagation and analysis

- **Detection is asynchronous**
  - at each event, not at sampled time, not continuously
  - it checks that the system has a certain dynamics under a certain control between two configuration changes

---

From discrete event to hybrid (5)

**Computing the time windows attached to transitions:**

- $t_1$: plan specifies between date$_{t_1}$ and date$_{t_1}$ - fires at date$_{t_1}$
- $p_3$: from physical device duration between dur$_{t_1}$ and dur$_{t_2}$
- $t_2$: plan specifies between date$_{t_2}$ and date$_{t_3}$
- after firing $t_1$ derive interval $[\text{date}_{t_1} + \text{dur}_{t_1}, \text{date}_{t_1} + \text{dur}_{t_2}]$
- if disjoint from $[\text{date}_{t_1}, \text{date}_{t_2}]$ then error (inconsistency of plan)
- if date$_{t_1}$ out of $[\text{date}_{t_1}, \text{date}_{t_2}]$ then plan lack of flexibility
- if date$_{t_2}$ out of $[\text{date}_{t_1} + \text{dur}_{t_1}, \text{date}_{t_1} + \text{dur}_{t_2}]$ then physical system failure
From discrete event to hybrid (6)

Computing the time windows attached to places:

- **For time events (output transition)**
  - \( p_t \): defined at the level of the production route, or recipe: stir during 5 minutes

- **For constant state events (output transition)**
  - \( p_c \): the reactor is empty when \( t_1 \) is fired, the batch size is constant, the flow rate is constant, duration is the same for all batches

- **For variable state events (output transition)**
  - \( p_v \): the temperature to be reached depends on the batches, duration has to be computed on line

Variable state events:

- interval depends on the dynamics of continuous variables
- necessity of using a hybrid model (PN+DAE) for supervision