ES623 Networked Embedded Systems

Modeling Real-Time Systems

Interfaces (contd)

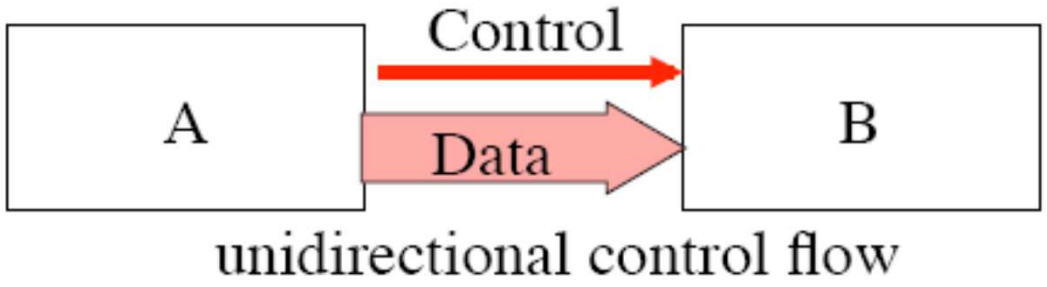
08th March 2013

Interfaces

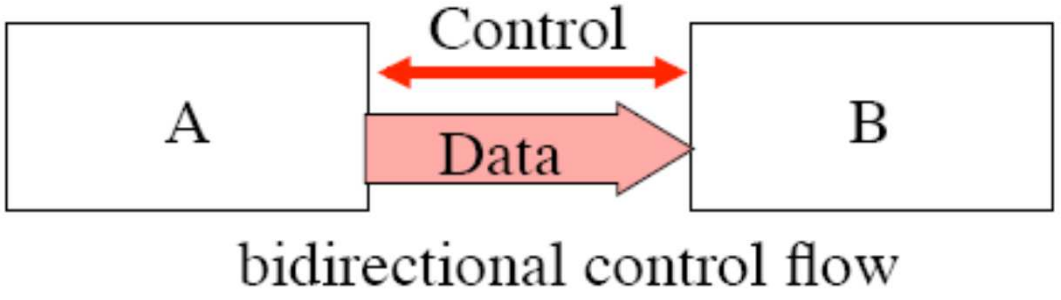
- Common boundary between two subsystems
- Characterized by
  - ***data properties***, i.e., the structure and semantics of the data items crossing the interface.
  - The semantics include the ***functional intent***, i.e., the assumptions about the functions of the interfacing partner
  - Its ***temporal properties***, i.e., the temporal conditions that have to be satisfied by the interface: e.g., update rate and temporal data validity
  - Its ***control properties***, i.e., strategy used to control the data transfer between reader and writer

Elementary Vs Composite interface

Elementary Interface:

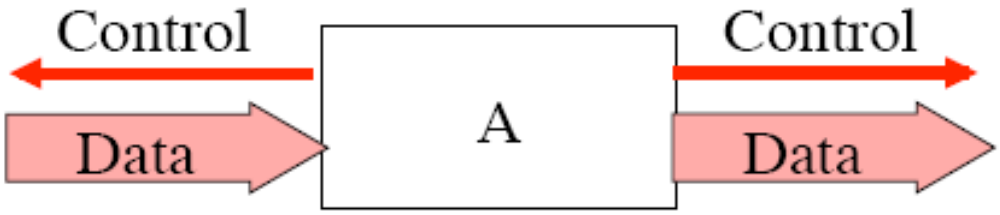


Composite Interface:



Temporal-Firewall

- Interface that does not allow to execute external control over the component



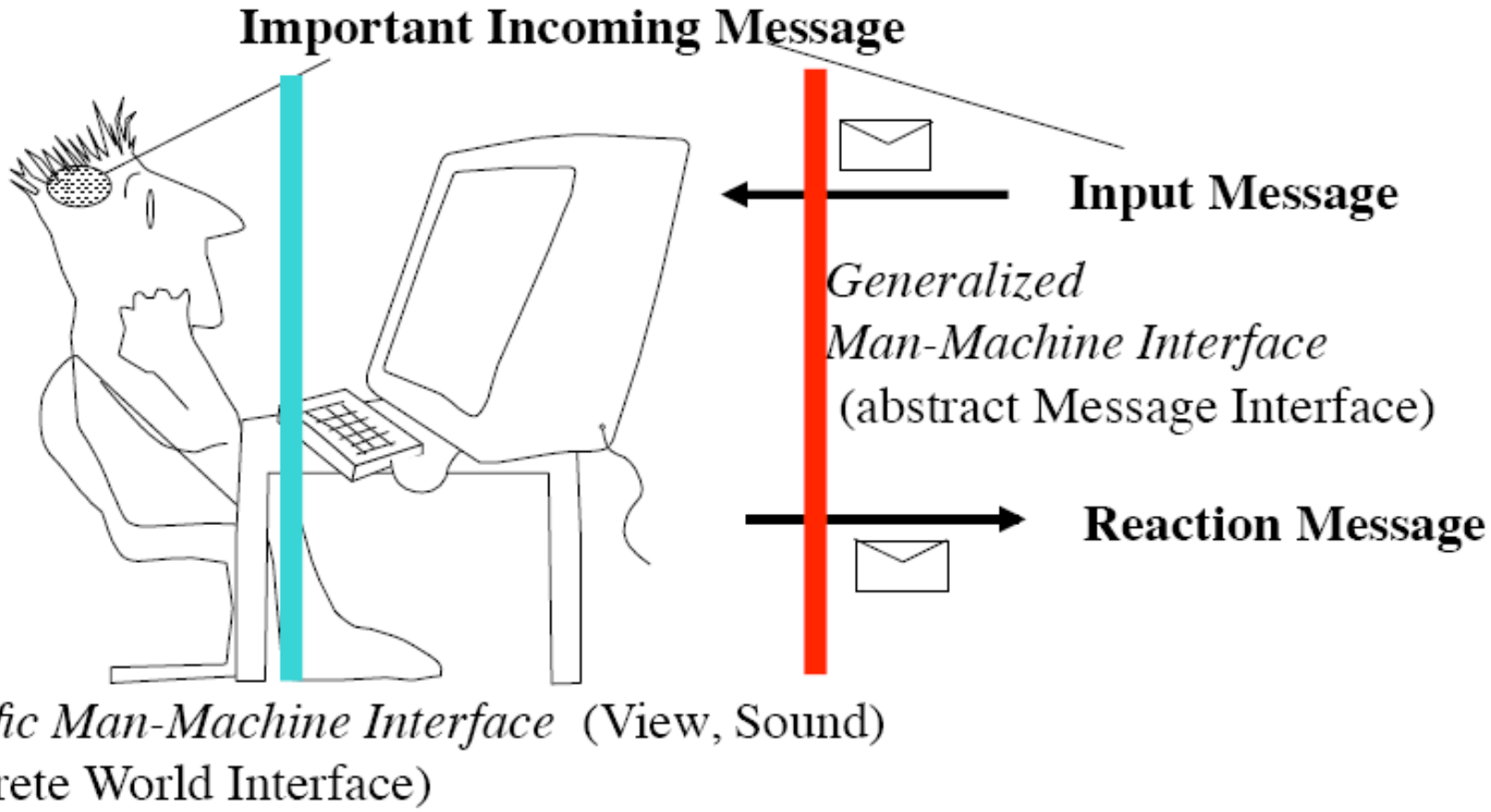
World and Message Interfaces

- low level interface the *world interface*
- internal abstract message-based the *message interface*
- interface component between the message and the world interface acts as an “information transducer” and is called *resource controller*

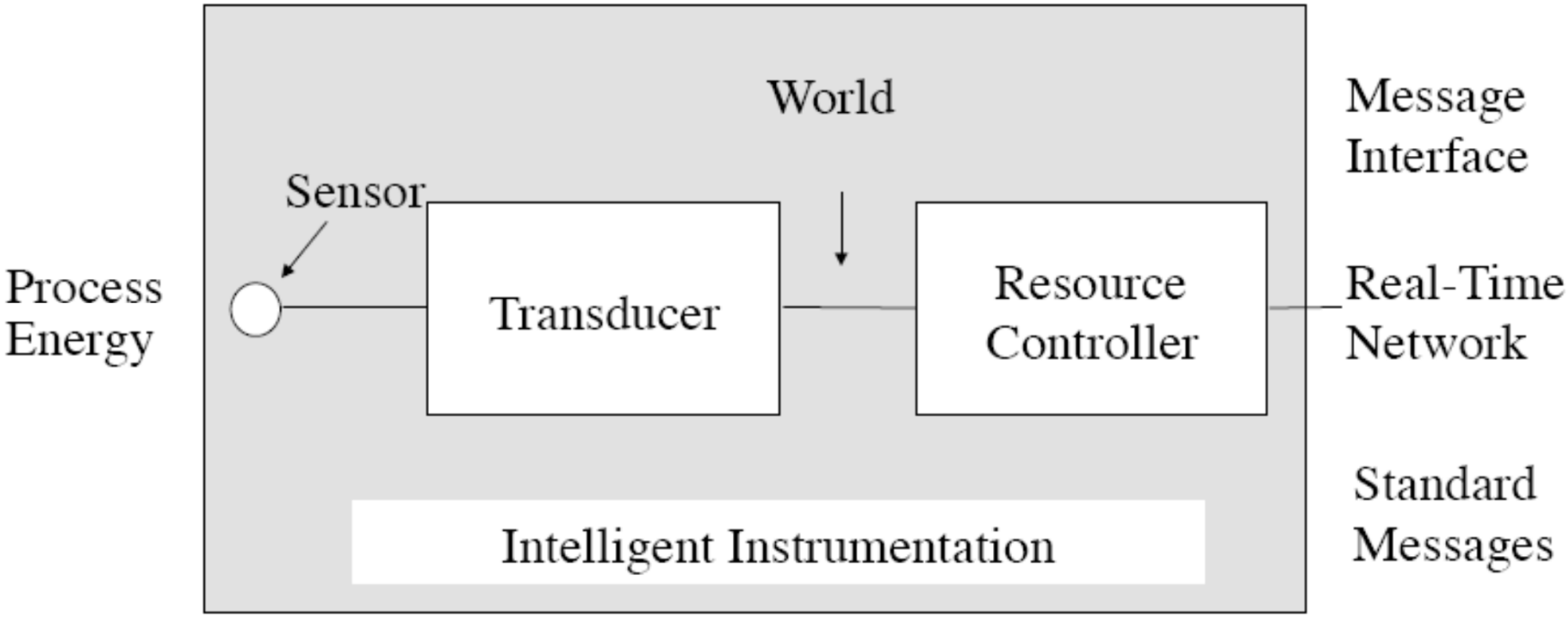
World and Message Interfaces

- Man-Machine interface (MMI)
  - *Specific man-machine interface (SMMI)*
  - *Generalized man-machine interface (GMMI)*
- SMMI
  - Concrete world interface
  - between machine and human operator
- GMMI
  - Abstract message interface
  - between MMI and rest of distributed system

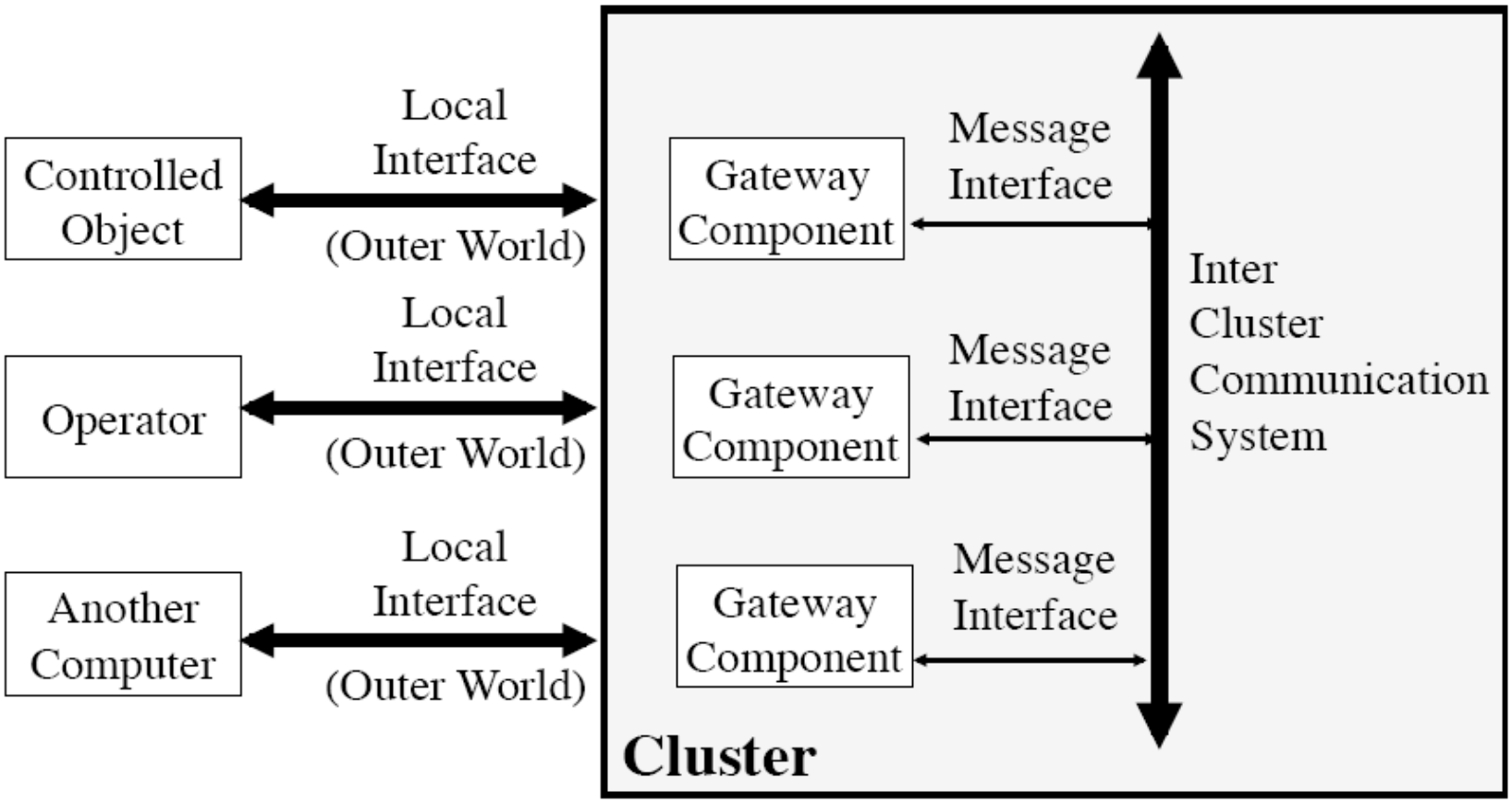
Example: Man-Machine Interface



Example: Intelligent Interface



World and Message interface in distributed system



Temporal obligations of clients & servers

- In client-server model, a request (a message) from a client to a server causes a response from the server
- The response could be a state change of the server and/or the transmission of a response message to the client
- Client-server interaction characterized by three temporal parameters
  - RESP
  - WCET
  - MINT

Client – server interaction

- Maximum response time, RESP, that is expected by the client, and stated in the specification
- Worst-case execution time, WCET, of the server, determined by the implementation of the server
- Minimum time, MINT, between two successive requests by the client
- WCET – sphere of control of the server
- MINT – sphere of control of the client
- In hard real-time environment,

$$WCET < RESP$$

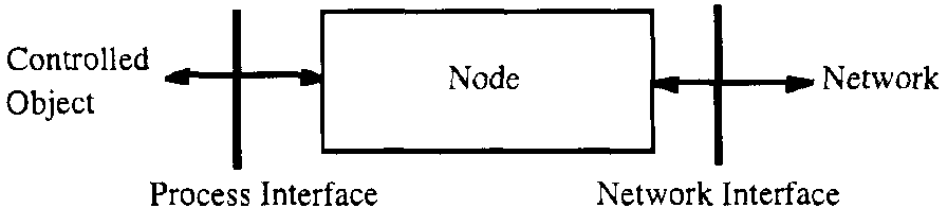
Under the assumption that MINT is minimum

Client – server interaction

- If WCET << RESP holds, performance of the server is faster than the required by particular application under worst-case conditions – hardware 'over dimensioned'
- Exceptional condition due to cost pressure in market
- WCET & RESP of same magnitude, server meet the temporal requirements, provided the client issues its requests only at a rate less than 1/MINT

Client – server interaction

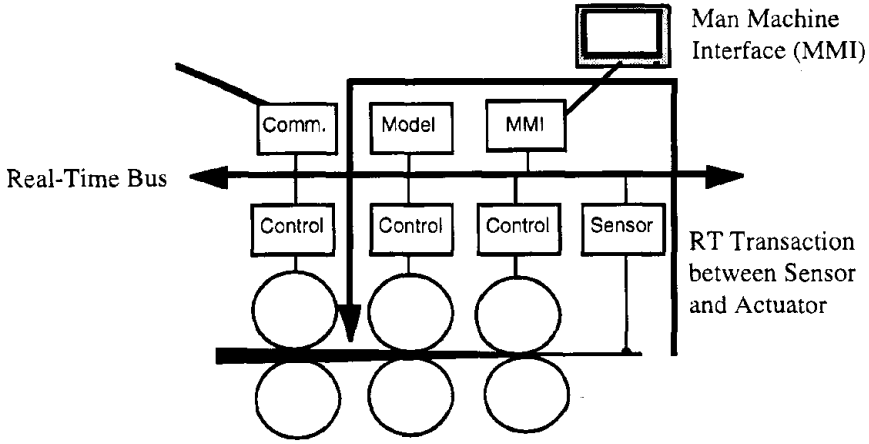
- Node of distributed system has two interfaces
  - provides services to the network across Network interface
  - provides services to the controlled object across Process interface



- If the service activation is not in the sphere of control, timely operation of the node is not possible
- Results in consequent failure of the node

Temporal Control Versus Logical Control

- Consider a Rolling Mill Example
  - Variables are measured and monitored by the alarm monitoring unit
- Assume that pressure *p1*, *p2*, and *p3* between rolls of three drives are measured by three controller nodes and sent to MMI node to check the following condition



when *((p1<p2)^(p2<p3))*

then *everything ok*

else *raise pressure alarm;*

Temporal Control Versus Logical Control

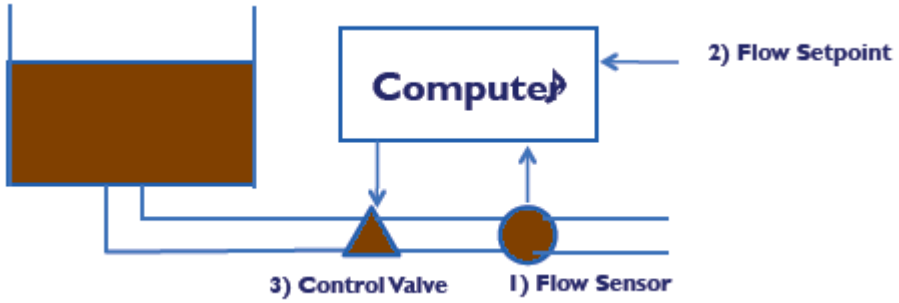
- *Logical control* is concerned with the *control flow within a task* that is determined by the given program structure and particular input data, in order to achieve desired data transformation
- *Temporal control* is concerned with determining the *points in time* when a task must be activated or when it must be blocked, because conditions outside the task are not satisfied at a particular moment
- Only temporal control issue in s-task is determining the moment
- C-task blends issues of logical with temporal control

Real-Time Entities

11th March 2013

Real-Time Entities

- *Real-time (RT) entity is a state variable of relevance for the given purpose*
- located either in the environment or in the computer system.
- Examples of RT entities:
  - the flow of a liquid in a pipe
  - the set point of a control loop
  - the intended position of a control valve

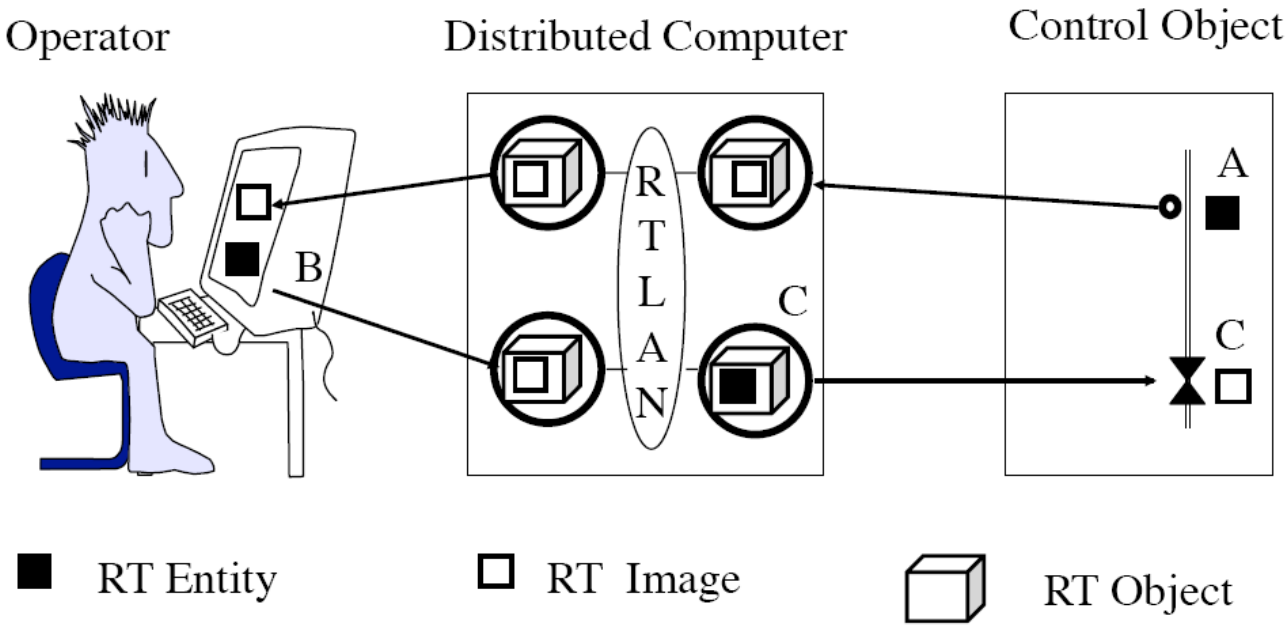


Real-Time Entities

- *RT Entity has*
  - Static attributes – that do not change during the lifetime
    - Examples: name, type, value domain, maximum rate of change
  - Dynamic attributes – that change with time
    - Examples: value set at a particular point in time, rate of change at a chosen point in time
- Every RT entity is in the sphere of control (SOC)
- Outside SOC RT entity can only be observed, not modified

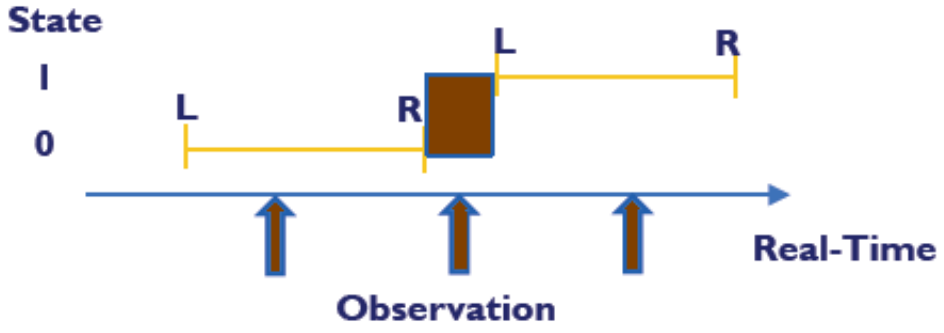
Real-Time Entities

- *Three RT Entity are:*
  - Set-point is in the SOC of the operator
  - Actual Flow is in the SOC of the control object
  - Intended Valve Position is in the SOC of the Computer



Real-Time Entities

- Continuous RT Entity
  - The set of values is always defined
  - Example: Flow in a Pipe
- Discrete RT Entity
  - Have a discrete value set which remains constant between ***a left event (L_event) and a right event (R_event)***
  - In the interval between an R_event and the next L_event, the set of values is ***undefined***
  - Example: Position of a Switch



RT Observations

- The role of observations:
  - To capture the information about the state of an RT entity at a particular point in time

$$Observation = <Name,\ t_{obs},\ Value>$$

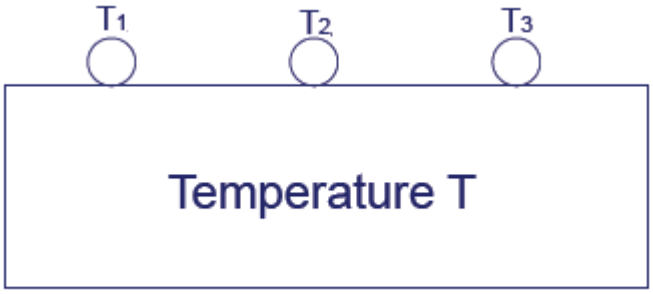
  - *Name of the entity,*
  - t_{obs} *point in real time when observation was made*
  - $Value$ *Observed value of the RT entity*
- Classification of Observations
  - Untimed Observation
  - Indirect Observation
  - State Observation
  - Event Observation

RT Observations

- Untimed Observation
  - Without global time, a timestamp generated by the sender is meaningless at the receiver.
  - Receiver use ***the time of arrival of the untimed*** observation message to decide t_{obs}
  - This t_{obs} ***may be imprecise (delay and the jitter)***

RT Observations

- Indirect Observation
  - Example: the measurement of the temperature within a slab of steel.
  - May need a mathematical model of heat transfer

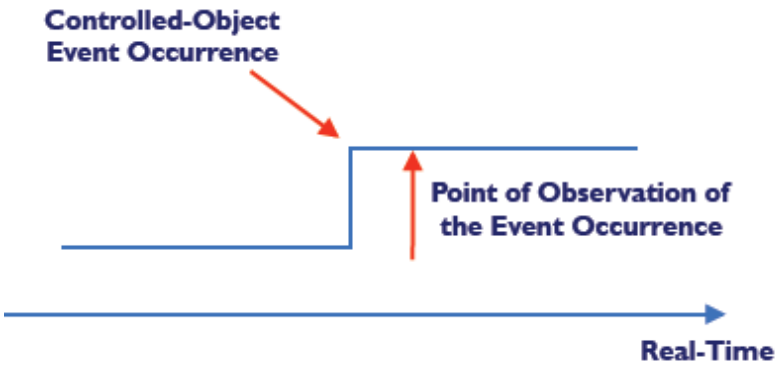


RT Observations

- State observation *Observation = <Name, t_{obs} , Value>*
  - *Value : the state of the RT entity*
  - t_{obs} *: point in real-time when the RT entity was* sampled
  - Every reading of a state observation carries an *absolute value*

RT Observations

- Event Observation *Observation = <Name, t_{obs}, Value>*
  - *Value : the change in value between the "old" and the "new" states*
  - t_{obs} *: the best estimate of the point in time of the event*



Fault Tolerance

12th March 2013