EXAMPLES OF D.E.S.

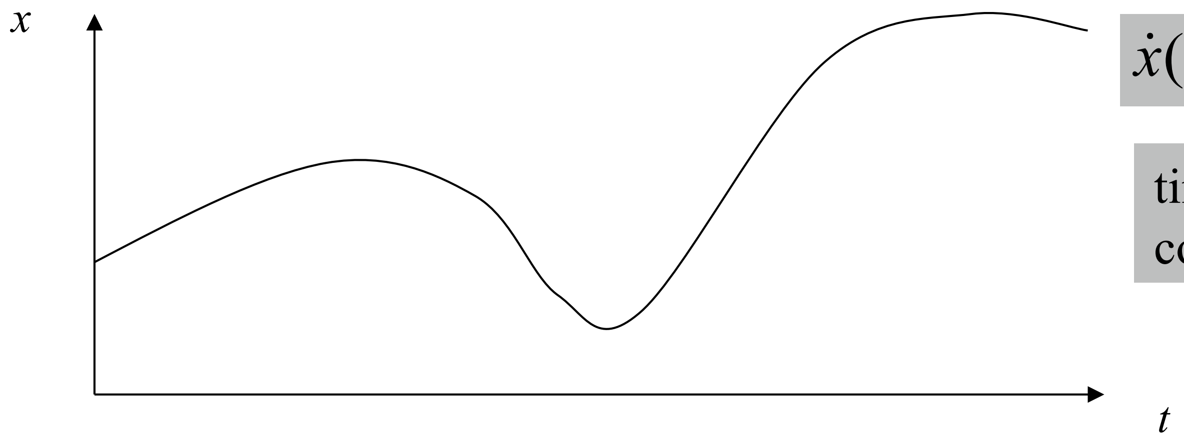
Pedro U. Lima

Instituto Superior Técnico (IST)
Instituto de Sistemas e Robótica (ISR)
Av.Rovisco Pais, 1
1049-001 Lisboa
PORTUGAL

September 2002

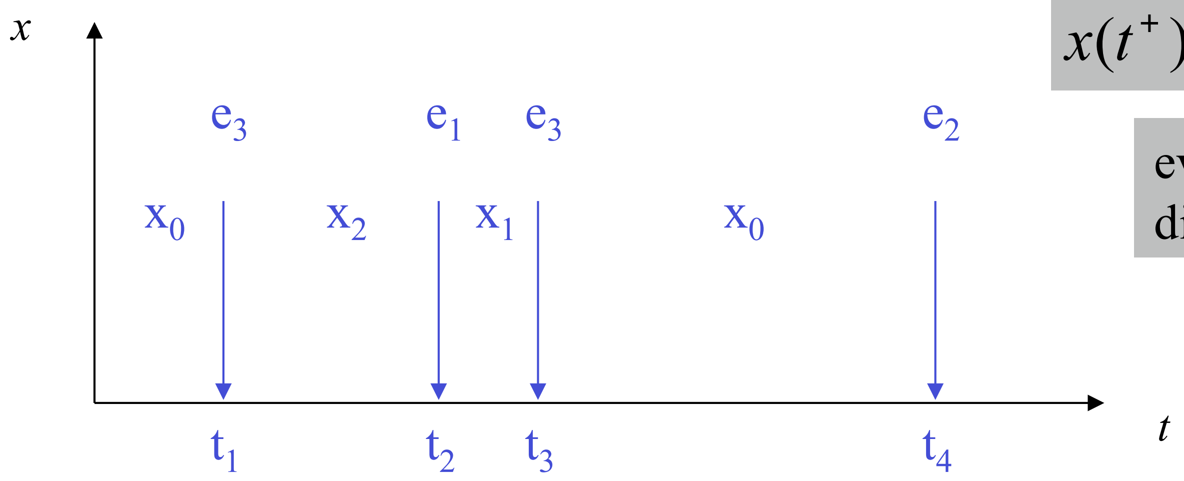
INSTITUTO SUPERIOR TÉCNICO

DISCRETE EVENT DYNAMIC SYSTEMS



$$\dot{x}(t) = f(x(t), u(t), t)$$

time-driven
continuous state space



$$x(t^+) = f(x(t^-), u(t^-))$$

event-driven
discrete state space

$$u(t) = e_3 e_1 e_3 e_2$$

DISCRETE EVENT DYNAMIC SYSTEMS

State of a system at time t_0 is the information required at t_0 such that the output $\mathbf{y}(t)$, for all $t \geq t_0$, is uniquely determined from this information and from $\mathbf{u}(t)$, $t \geq t_0$.

State Space - Discrete set *X*

Event is a primitive concept, intuitively understandable, occurring instantaneously and causing transitions from one state value to another (e.g., a pressed button, a failure, reaching a threshold, detecting an object)

Events - Discrete set *E*

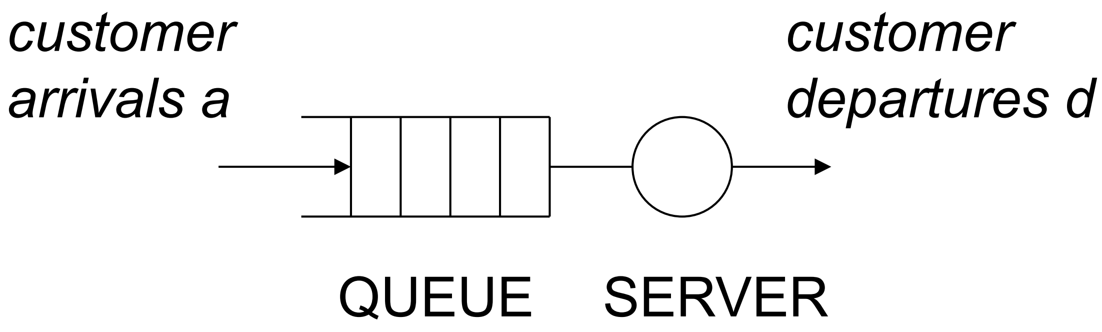
DISCRETE EVENT DYNAMIC SYSTEMS

The **three levels of abstraction**:

- ***(untimed) language***: set of all possible ordering of events that could happen in a given system
- ***timed language***: set of all timed sequences of events that a given system can ever execute
- ***stochastic timed language***: a timed language with associated probability distribution functions (pdf) for the events

Level of Abstraction	Analysis Objectives
Untimed language	study/prescribe logical behavior (e.g., a precise ordering of events), avoid unsafe state(s)
Timed language	how long is the system in a given state?, can this sequence of events be accomplished before the deadline?
Stochastic timed language	reliability, probability of accomplishing a task

Example of DES – Queueing Systems

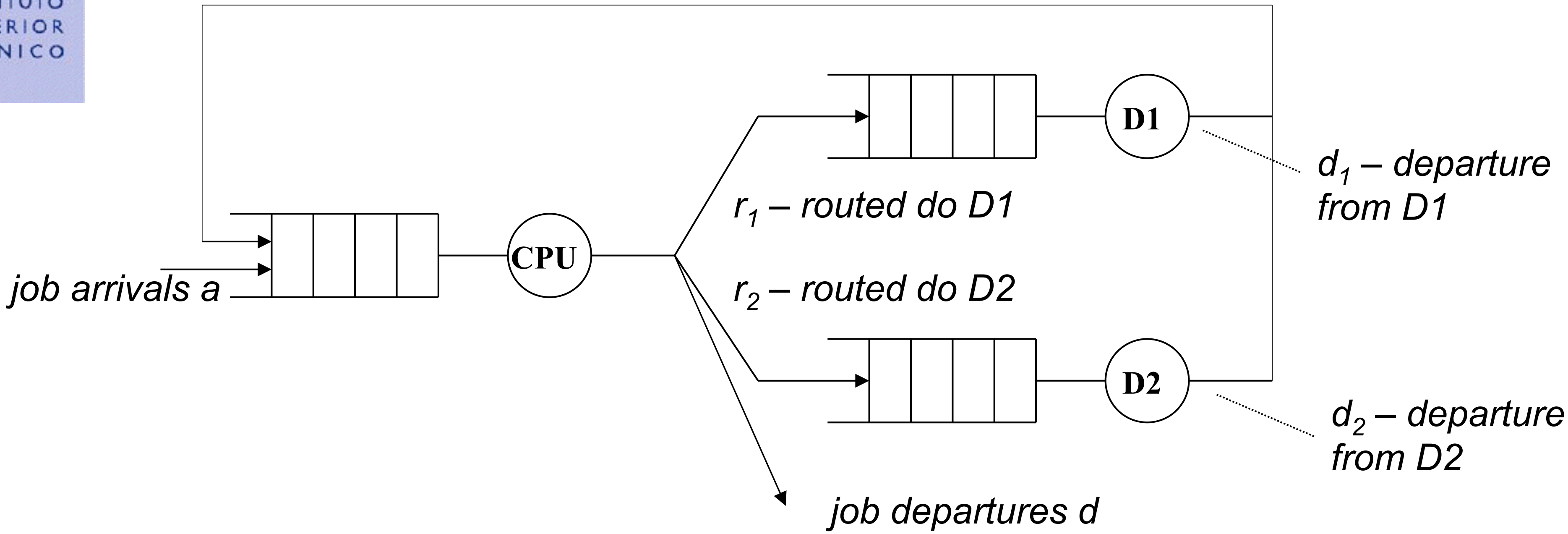


Customers: *Entities* that wait for availability of resources (e.g., people, parts to be machined, computer jobs, communication messages)
Servers: *Resources* which provide some form of service to the customers (e.g., counter tellers, manufacturing machines, computer processors, communication channels)
Queue: Space where the waiting is done (e.g, bank lobbies, bus stops, machine buffers, printer queue, channel buffer)

$E = \{a,d\}$

$X = \{0,1,2,\ldots\}$ – queue length (including customer being served)

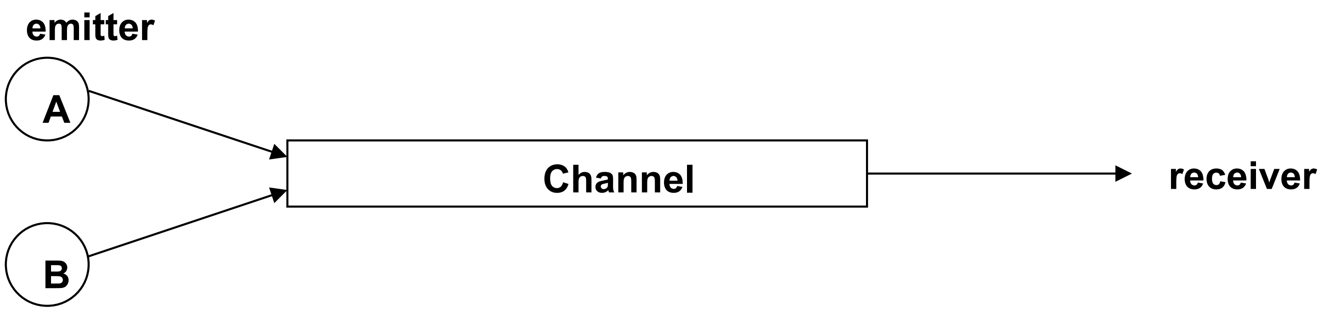
Example of DES – Computer Systems



$E = \{a, d, r_1, r_2, d_1, d_2\}$

$X = \{(x_{CPU}, x_1, x_2): x_{CPU}, x_1, x_2 \geq 0\}$ – **queue lengths at CPU and disks D1 and D2**

Example of DES – Communication Systems



Events:

- $a_{A,B}$ arrivals of msgs to be transmitted by A(B)
- $\tau_{A,B}$ sending a message by A(B)
- τ_{CH} completion of a channel transmission

Channel state:

- I idle
- T transmitting one message
- C collision (attempting to transmit > 1 msg)

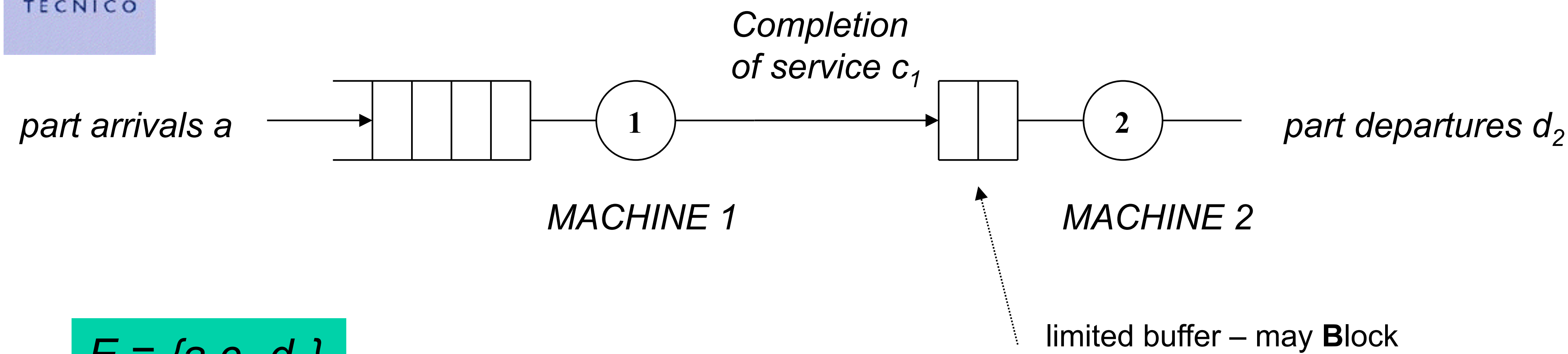
Emitter state:

- I idle
- T transmitting
- W waiting to transmit

$$E = \{a_A, a_B, \tau_A, \tau_B, \tau_{CH}\}$$

$$X = \{(x_{CH}, x_A, x_B)\}$$

Example of DES – Manufacturing Systems



$E = \{a, c_1, d_2\}$

$X = \{(x_1, x_2): x_1 \geq 0, x_2 \in \{0, 1, 2, 3, B\}\}$

OR

$X = \{(x_1, x_2): x_1 \in \{I, W, B\}, x_2 \in \{I, W\}\}$

Example of DES – Database Systems

In Databases, the sequence of reads and writes of an user is called a *transaction*.

A *schedule* is the sequence of events representing the actual order in which the operations of the transactions are executed on the database.

Database Management Systems (DBMS) must ensure database consistency in the presence of concurrent transactions (*concurrency control problem*).

EXAMPLE: banking transactions

$r_i(a)$	reading record *a* by transaction *i*
$w_i(b)$	writing record *b* by transaction *i*

transaction 1	$r_1(a)\ r_1(b)$	*to obtain the total amount of money in a and b*
transaction 2	$r_2(a)w_2(a)\ r_2(b)w_2(b)$	*to transfer 100 EUR from a to b*

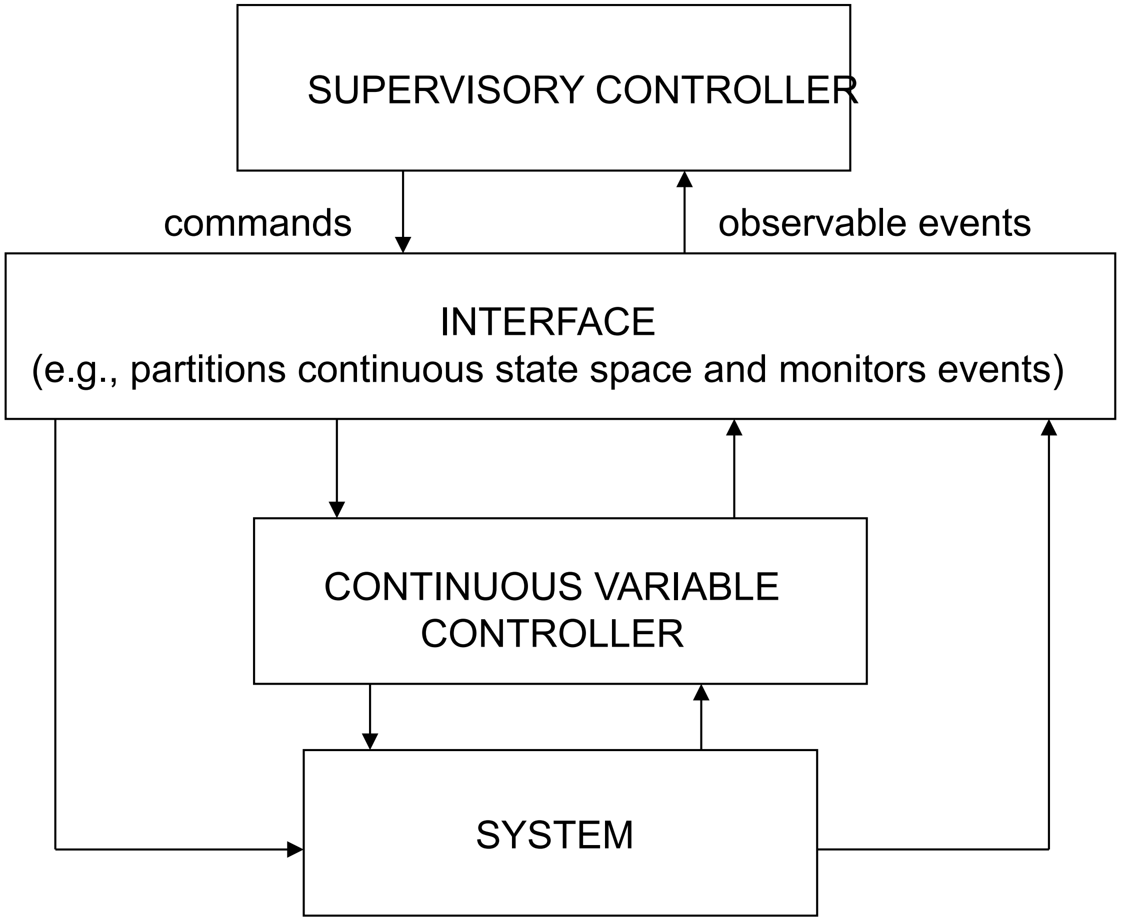
Schedules

$S_x = r_2(a)w_2(a)r_1(a)r_1(b)r_2(b)w_2(b)$ — transaction 1 returns 100 EUR less than what it should be

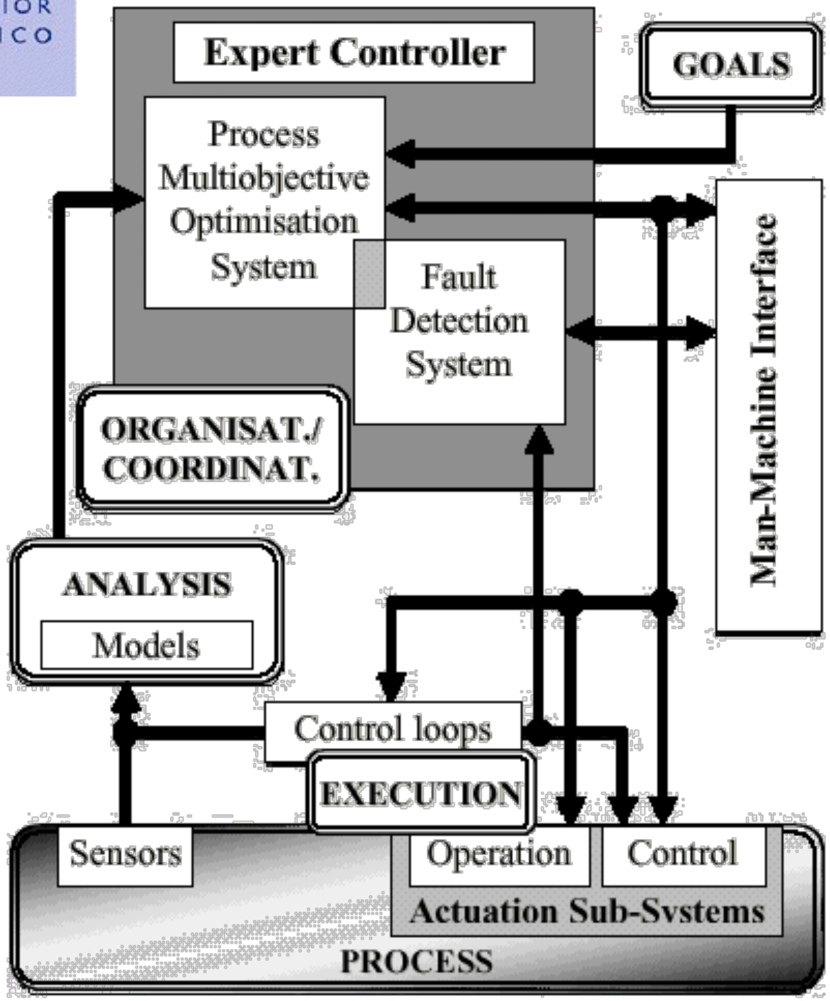
$S_y = r_2(a)w_2(a)r_1(a)r_2(b)w_2(b)r_1(b)$ — **OK**

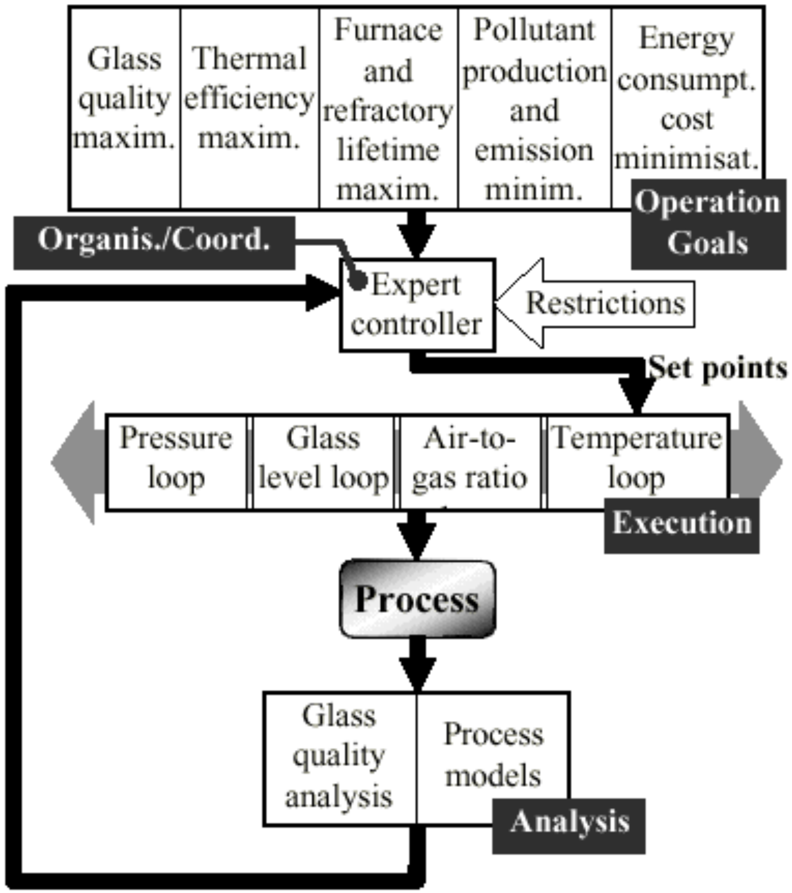
$S_z = r_1(a)\ r_2(a)w_2(a)r_2(b)w_2(b)$ — *problem!* Completing with $r_1(b)$ will lead transaction 1 to return 100 EUR more than the correct value – ***deadlock / blocking***

Example of DES – Process Monitoring and Operation



Example of DES – Process Monitoring and Operation



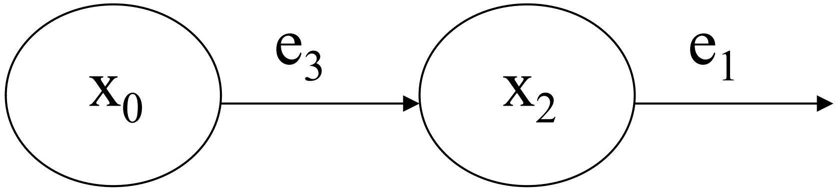


Example of DES – Robotic Systems

EXAMPLE: soccer robot

STATES	
x_0	move2ball
x_2	catchball

EVENTS	
e1	lostball
e3	nearball



implicitly associated:
- event monitoring
- primitive task algorithm