1 Introduction

1.1 Standard Model of a Real Time System (RTS)

1.2 Processes and Times

1.3 RTS in practice

Hints

Murphy's General Law

If something can go wrong, it will go wrong.

Murphy's Constant

Damage to an object is proportional to its value.

Naeser's Law

One can make something bomb-proof, not jinx-proof.

Troutman Postulates

1. *Any software bug will tend to maximize the damage.*
2. *The worst software bug will be discovered six months after the field test.*

Green's Law

If a system is designed to be tolerant to a set of faults, there will always exist an idiot so skilled to cause a nontolerated fault.

Corollary

Dummies are always more skilled than measures taken to keep them from harm.

Johnson's First Law

If a system stops working, it will do it at the worst possible time.

Sodd's Second Law

Sooner or later, the worst possible combination of circumstances will happen.

Corollary

A system must always be designed to resist the worst possible combination of circumstances.

/Buttazzo97/

What is realtime (RT)?

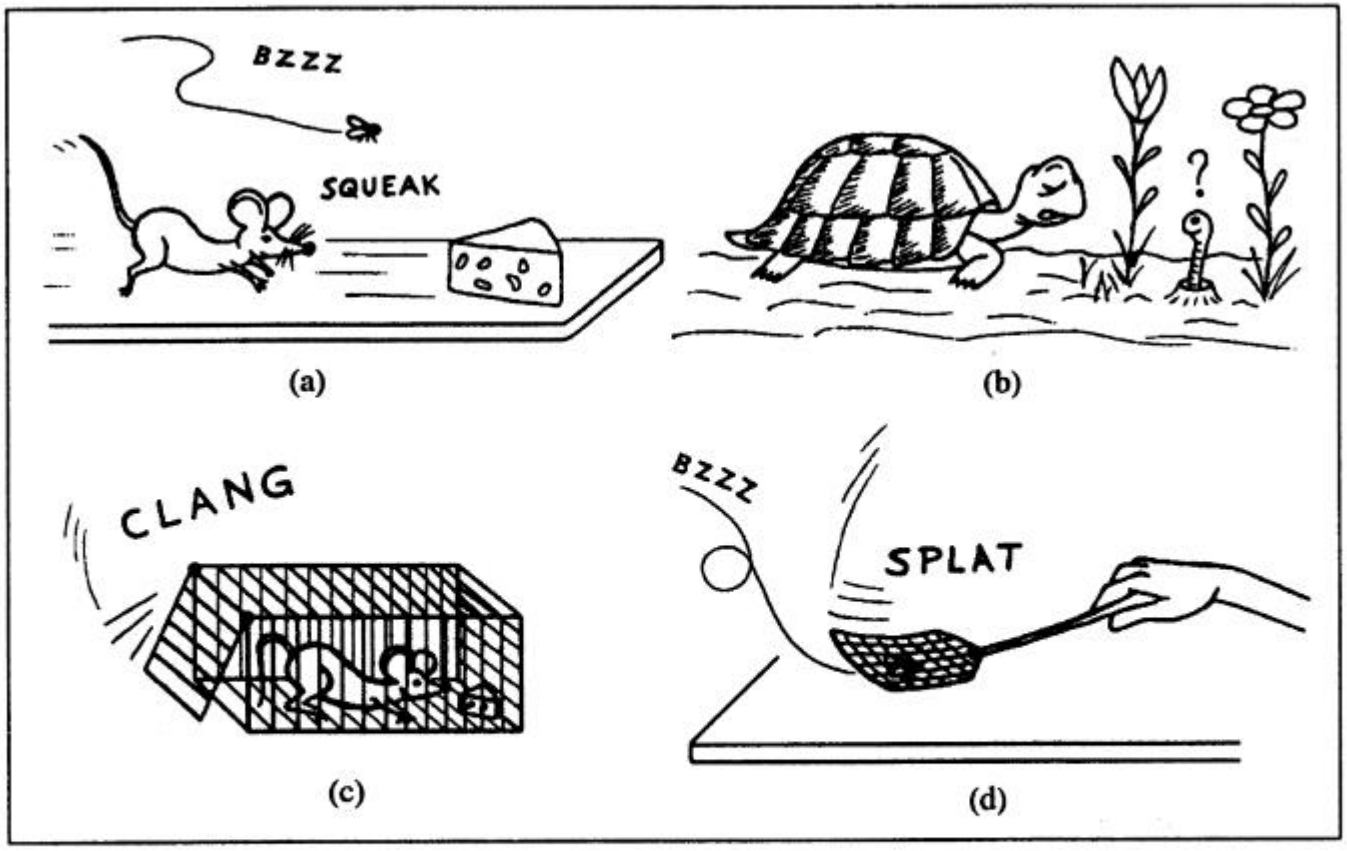


Figure 1.1 Both the mouse (a) and the turtle (b) behave in real time with respect to their natural habitat. Nevertheless, the survival of fast animals such as a mouse or a fly can be jeopardized by events (c and d) quicker than their reactive capabilities.

/Buttazzo97/

What is a realtime system (RTS)?

First try:
"RTS are very fast computer systems (HW+SW, hardware+software)."

First classification:
RST guarantees (always, average, most times, best effort) response Time r_T

r_T = { $10*10^{-6}$ s, $10*10^{-3}$ s, $100*10^{-3}$ s, 1s}

{ hard-, typical, normal, soft- } realtime

First examples:

- 3d following game
- airplane, fly by wire
- business data on the fly (SAP)
- airbag, ABS (anti blocking system), ESP (electronic stability packet), ...

Definition of the Standard

DIN 44300 (Deutsche Industrie-Norm, german industry norm
Deutsches Institut für Normung, german institute for norming)

Echtzeitbetrieb (Realzeitbetrieb):
Ein Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind, derart, daß die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind. Die Daten können je nach Anwendungsfall nach einer zeitlich zufälligen Verteilung oder zu vorherbestimmten Zeitpunkten anfallen.

"RT operation:
An operation of a computer system where the programs are always online so that the computational results are availabe within a given response time. The data can be input at random or defined times."

Hard and Soft RT conditions

Soft RT conditions:

- its sufficient, if the response times are kept in most of the cases
- the RT borders (e.g. the response time) may be crossed "a little"

Example:
At a travel agency the booking of a airplane seat shall be possible in 90% of all cases within 10s and in 99% of all cases within 20s.

Hard RT conditions:

- response times have to be kept in all cases
- RT borders are strict

Example:
An airbag has always to be ignited within 20ms after crash started. Exceptions are not accepted.

Hard RT conditions formal definition

Hard RT conditions: let

- r be the time, a task can be started (ready time)
- s the time, the task is started (starting time) (of course: s ? r)
- ? e the time the task needs for its execution (execution time*)
- d the task has to be ready (deadline)

than for the propability P one has to get

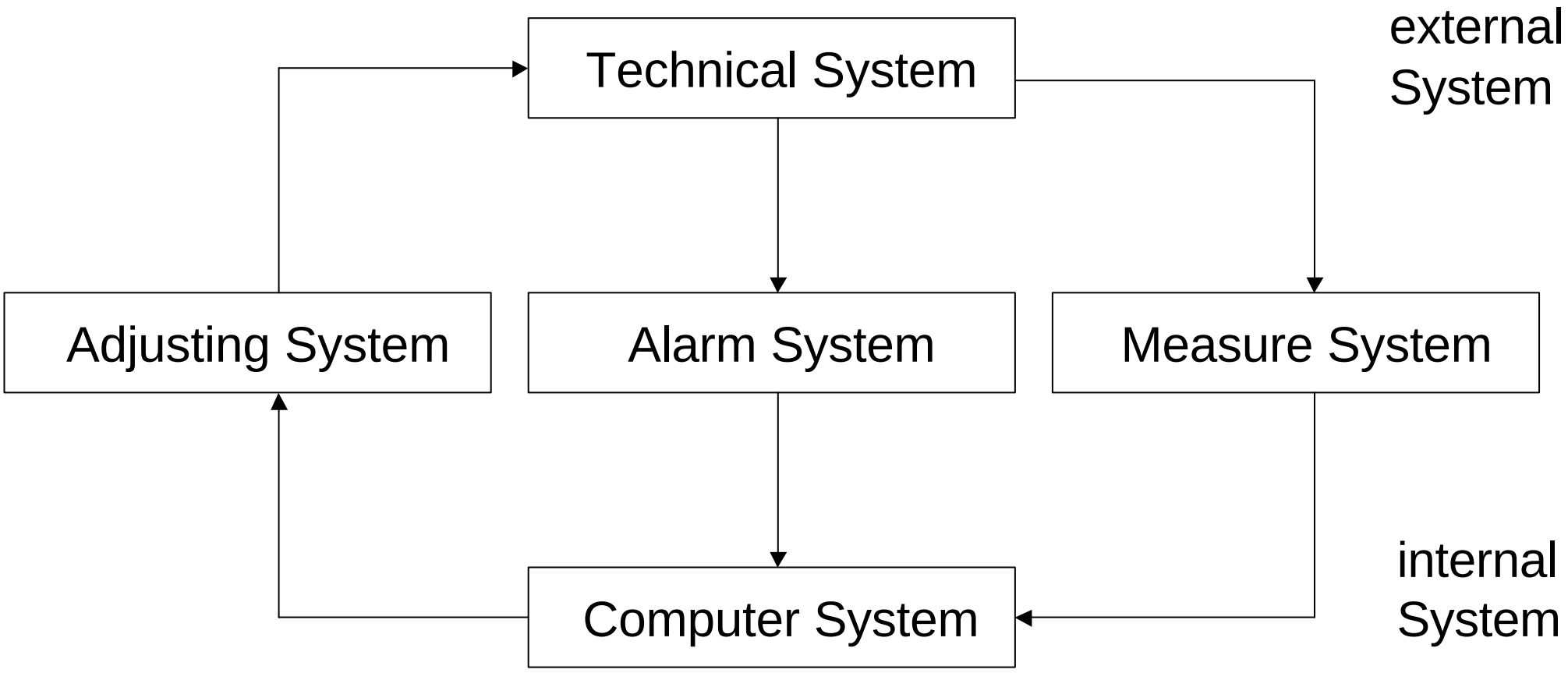
$$P(s + ?e ? d \mid B) = 1$$

so with a 100% probability under border conditions B times will be within their limits - the deadline is not violated.

* execution time is not only CPU time but also time for administration (e.g. scheduling, task changes etc.), waiting (e.g. bus) etc. (see embedded systems).

A RTS has to be fast but most important is its **predictability**.

First Model


Technical System
external System
Adjusting System
Alarm System
Measure System
Computer System
internal System

First model - similar approach

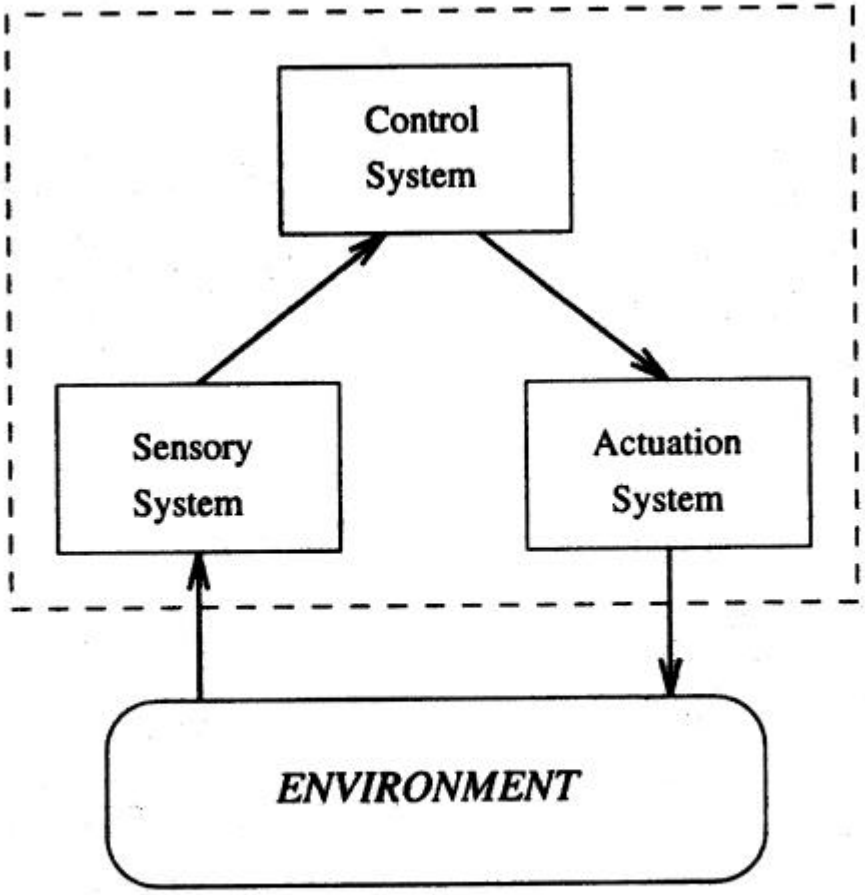
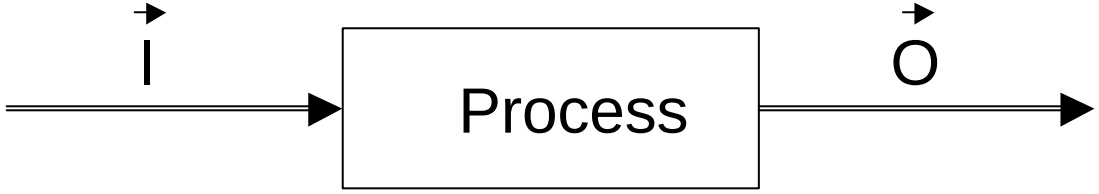


Figure 1.2 Block diagram of a generic real-time control system.

/Buttazzo97/

Process Definition



A process transforms an input vector into an output vector.

Model is simplified

Technical systems are very complex. So the corresponding RTS consist of many components - most of them hard to be described by the standard model.

Its hard to handle the system idea. The developers have to know the overall structure and they build subsystems separated by each other ("divide et impera").

The standard model implies a cyclical order of the sub steps but often steps can be done in parallel.

The knowledge of (sub)systems and their components is not complete.
This is right not only for the technical system but also for many SW components a RTS is built of.

So real RTS mostly neither fulfill hard RT conditions nor are they predictable!

1 Introduction

1.1 Standard Model of a Real Time System (RTS)

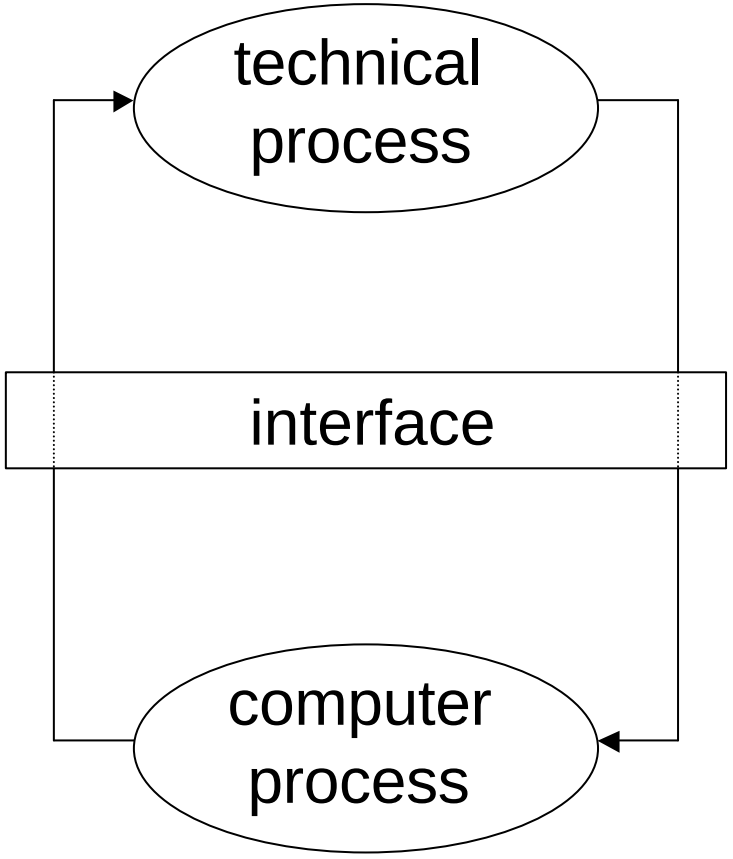
1.2 Processes and Times

1.3 RTS in practice

Processes and Times

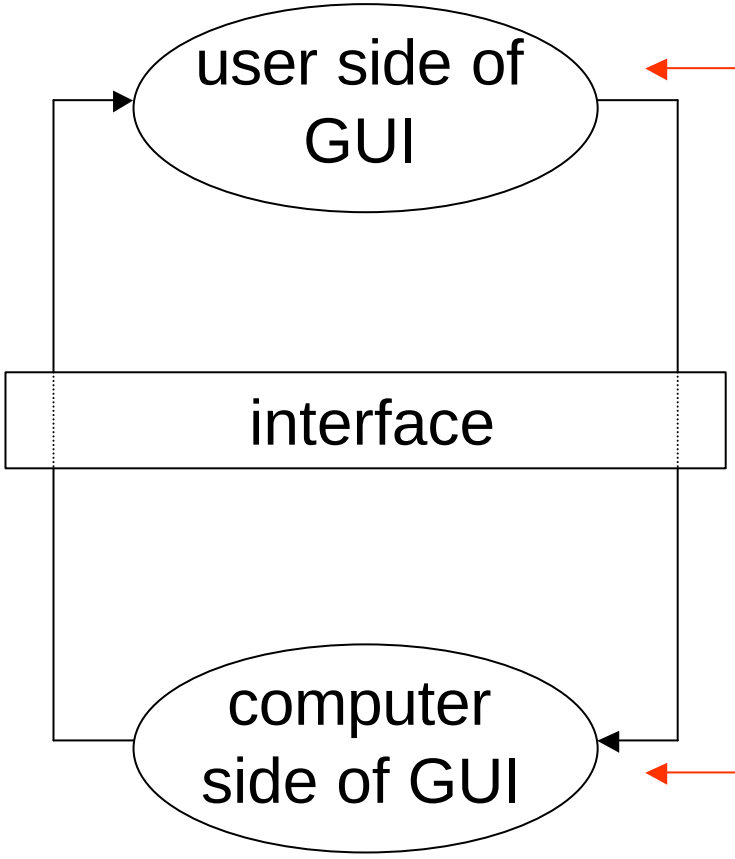
(Computional) process:
- center of activity in a (computer) system
- with a limited number of actions
- with results on a limited state space

First modeling:
- one computional process for
- one technical process



Standard model of a RTS with processes. Arrows indicate activity directions.

Example: Mouse and Screen

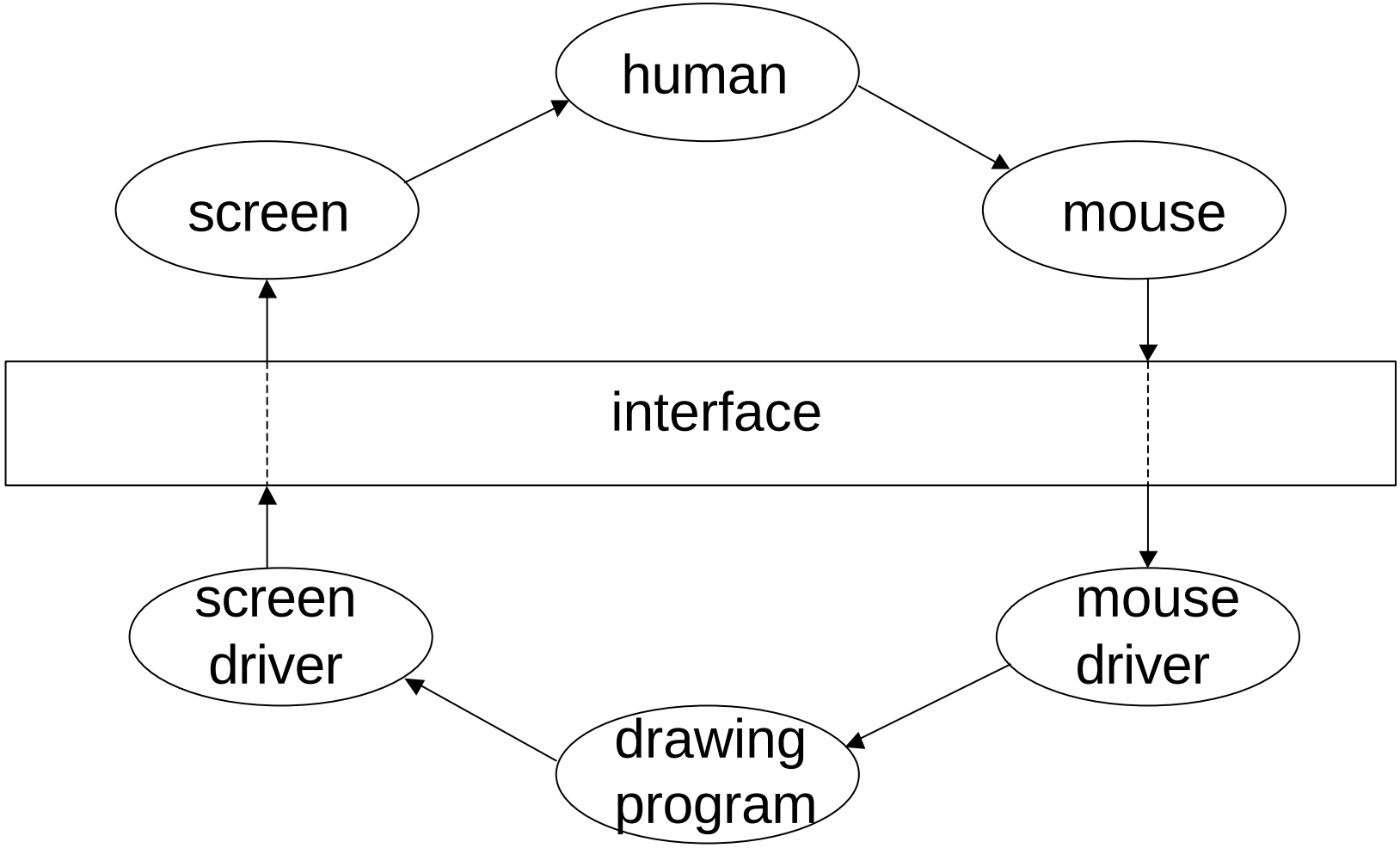


user! (human)
mouse
screen

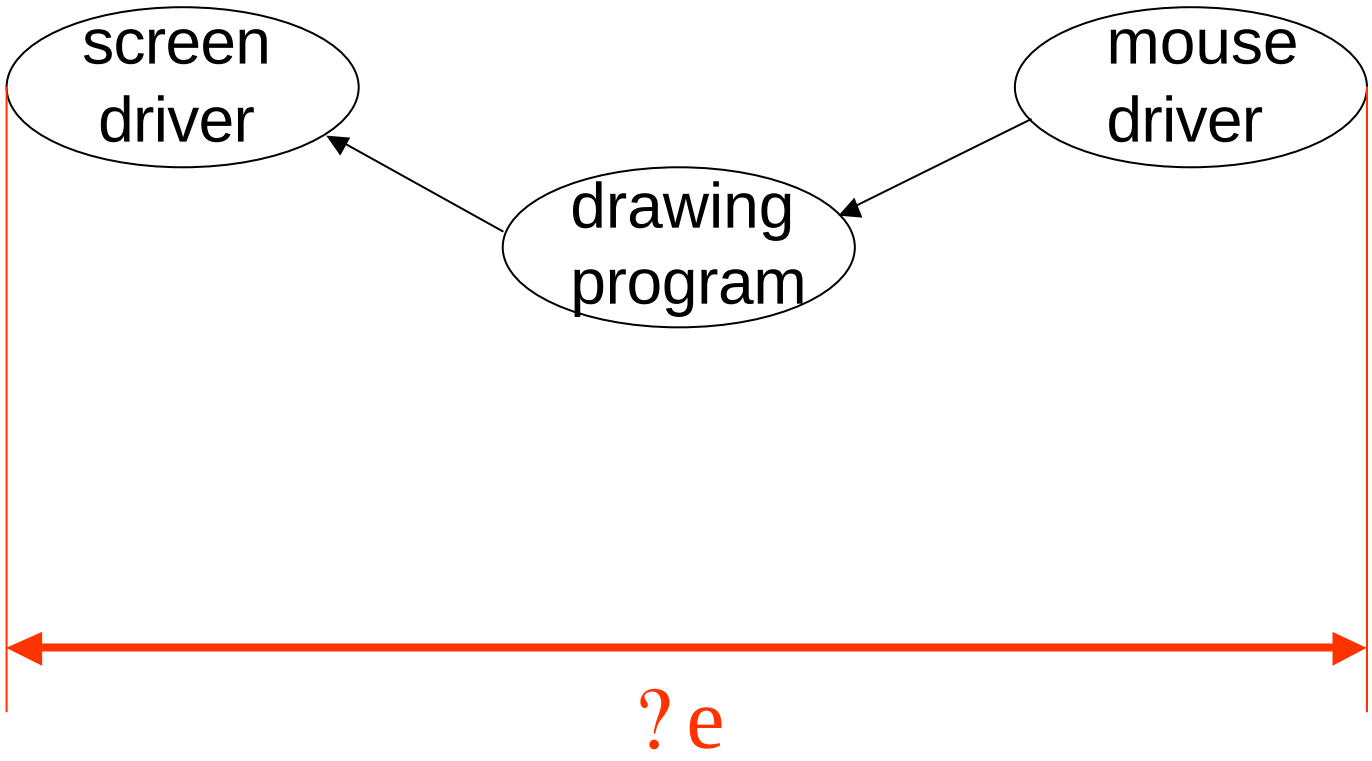
drawing program
mouse driver
screen driver

GUI: Graphical User Interface

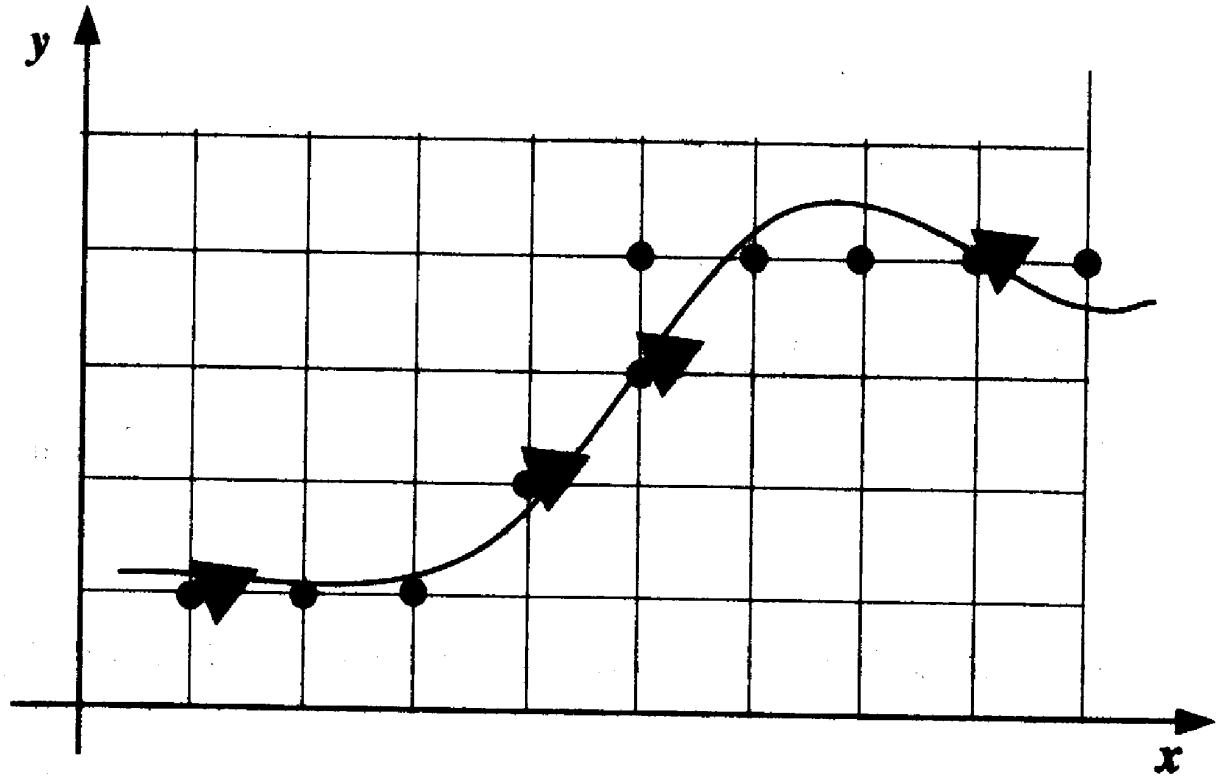
Example with Higher Granularity



Total time of computational process

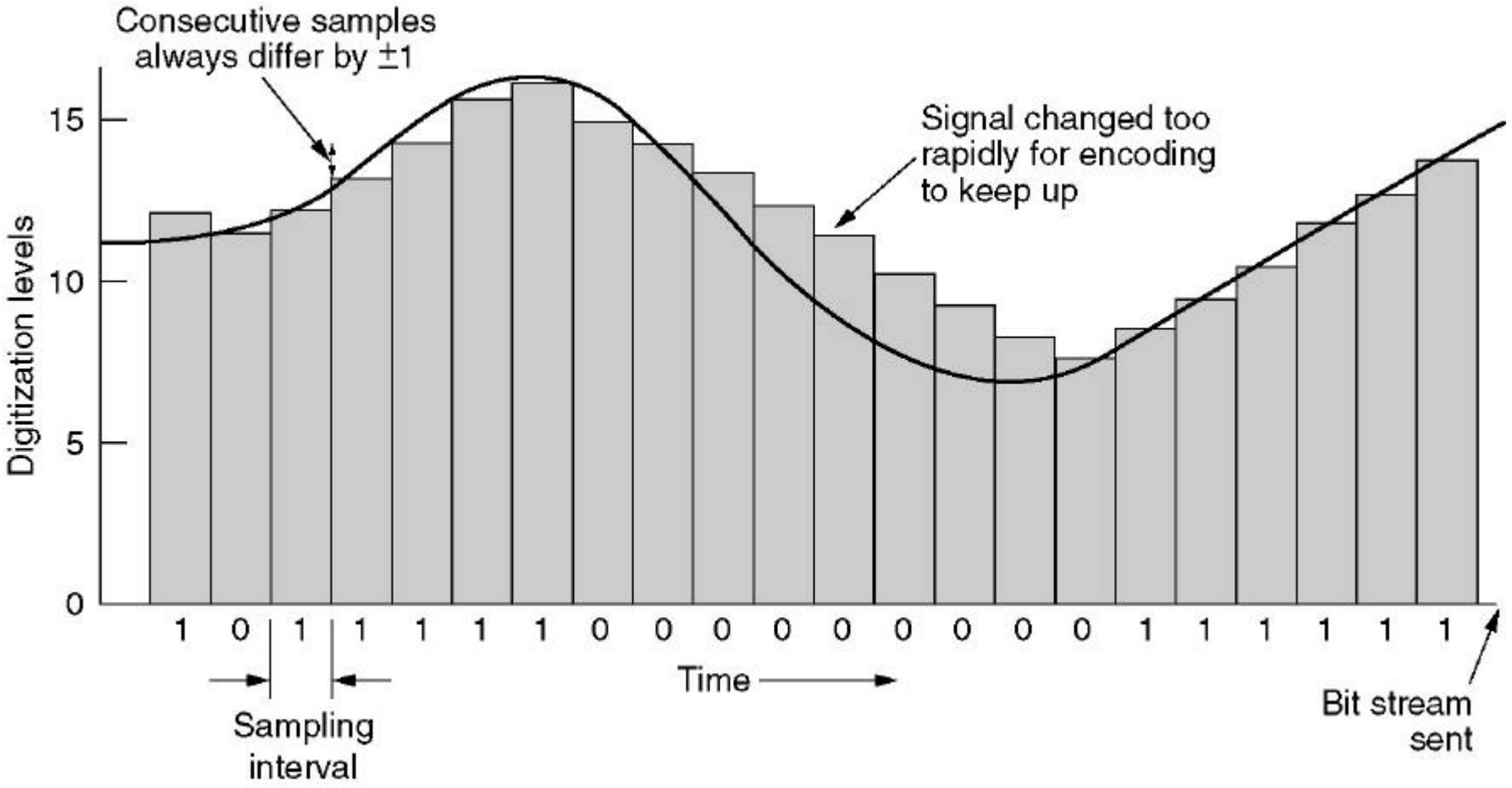


Computers are always discrete



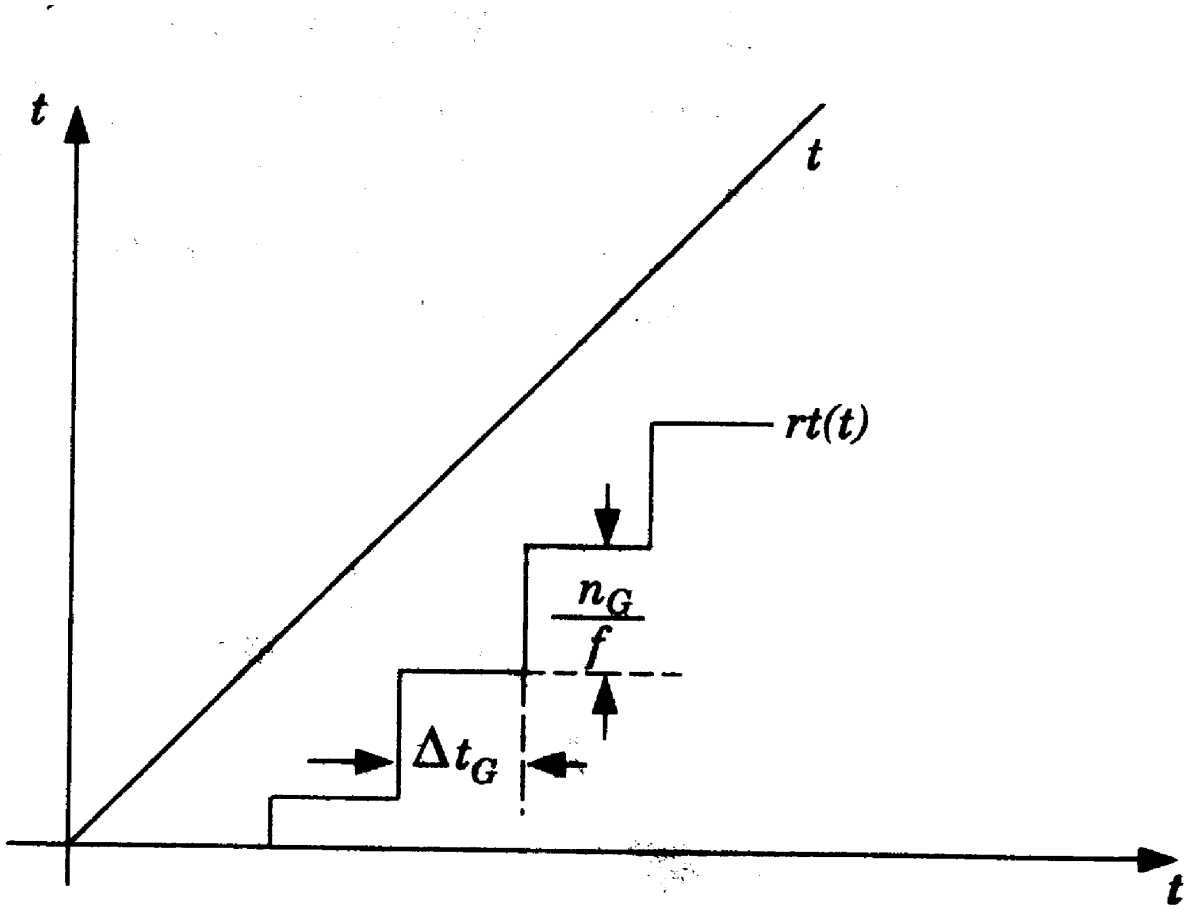
Physical movement of the cursor as a line plus translation
into discrete x- and y-values;
shown are are only some of these positions (see cursor symbol)

Discretisation causes errors



ADPCM, Delta Modulation

Discretional vs. physical time



Real time rt depending on physical time t

1 Introduction

1.1 Standard Model of a Real Time System (RTS)

1.2 Processes and Times

1.3 RTS in practice

Classification of RTS faults

RTS faults may cause
•loss of lives
•loss of money
•loss of quality
this defining a hierarchy.

Example:
within a car processing power is needed for ABS and the motor control (i.e. sparcs in every cylinder at the right time to get most power, least exhaust fumes, least gas consumption, least wear and tear etc.).
In a critical situation ABS has always to get sufficient processing power, even if this may cause a non optimal motor control.

RTS faults can be expensive

As an example, the first flight of the space shuttle was delayed, at considerable cost, because of a timing bug that arose from a transient CPU overload during system initialization on one of the redundant processors dedicated to the control of the aircraft. Although the shuttle control system was intensively tested, the timing error was never discovered before. Later, by analyzing the code of the processes, it has been found that there was only a 1 in 67 probability (about 1.5 percent) that a transient overload during initialization could push the redundant processor out of synchronization. /Buttazzo97/

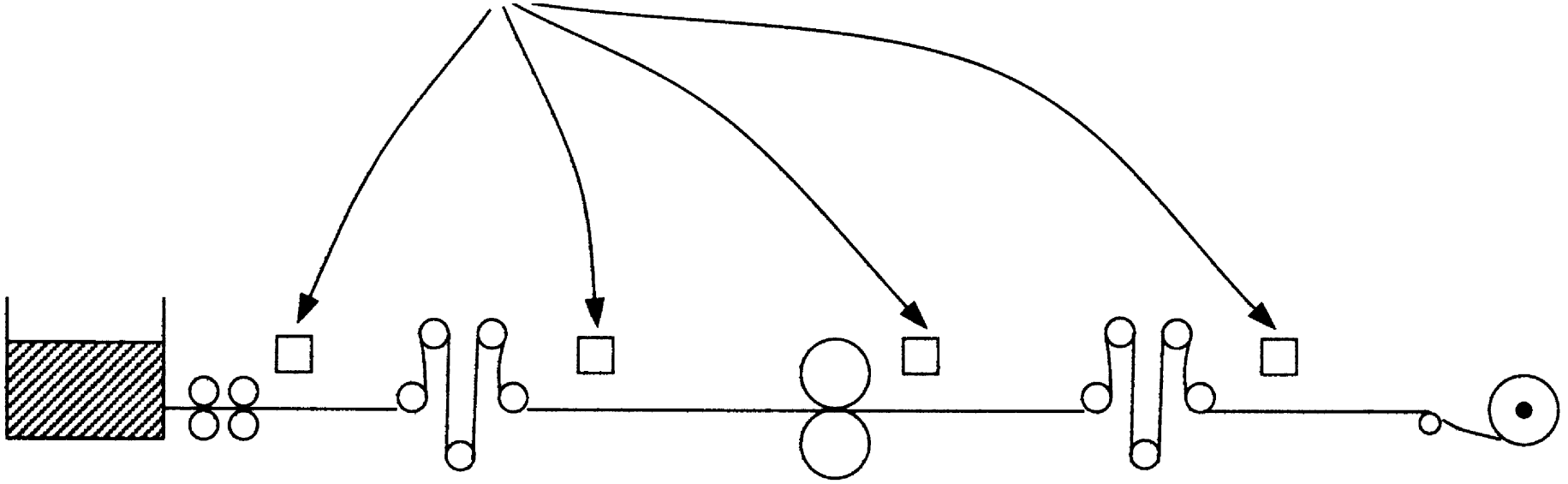
RTS faults can kill

Another software bug was discovered on the real-time control system of the Patriot missiles, used to protect Saudi Arabia during the (first) Gulf War. When a Patriot radar sights a flying object, the on-board computer calculates its trajectory and, to ensure that no missiles are launched in vain, it performs a verification. If the flying object passes through a specific location, computed based on the predicted trajectory, then the Patriot is launched against the target, otherwise the phenomenon is classified as a false alarm.

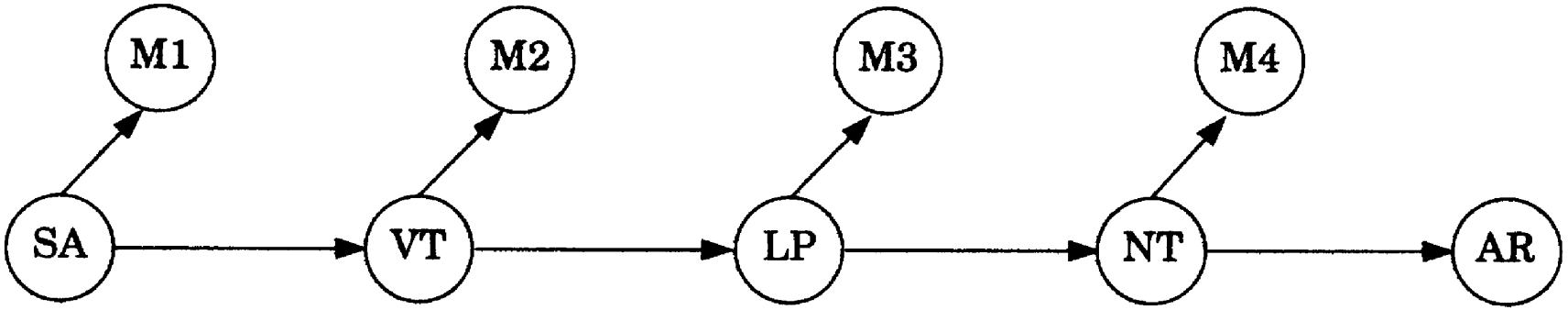
On February 25, 1991, the radar sighted a Scud missile directed at Saudi Arabia, and the on-board computer predicted its trajectory, performed the verification, but classified the event as a false alarm. A few minutes later, the Scud fell on the city of Dhahran, causing victims and enormous economic damage. Later on, it was discovered that, because of a subtle software bug, the real-time clock of the on-board computer was accumulating a delay of about 57 microseconds per minute. The day of the accident, the computer had heen working for about 100 hours (an exceptional condition that was never experienced before), thus accumulating a total delay of 343 milliseconds. This delay caused a prediction error in the verification phase of 687 meters! The bug was corrected on February 26, the day after the accident.
/Buttazzo97/

Example: paper producing machine, scheme

Measure Points (M1..M4) for thickeness, humidity etc.

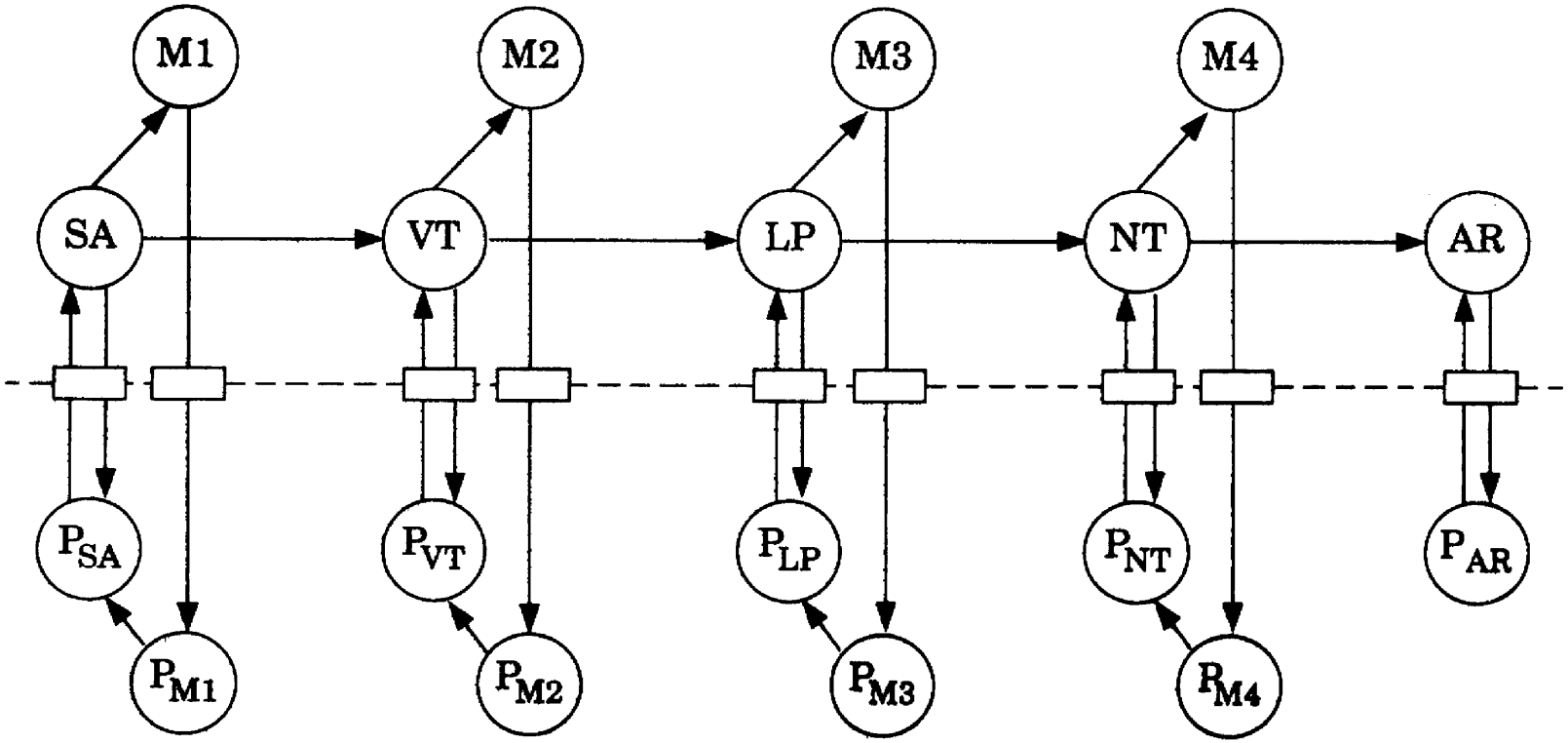


Example: paper producing machine, technical processes

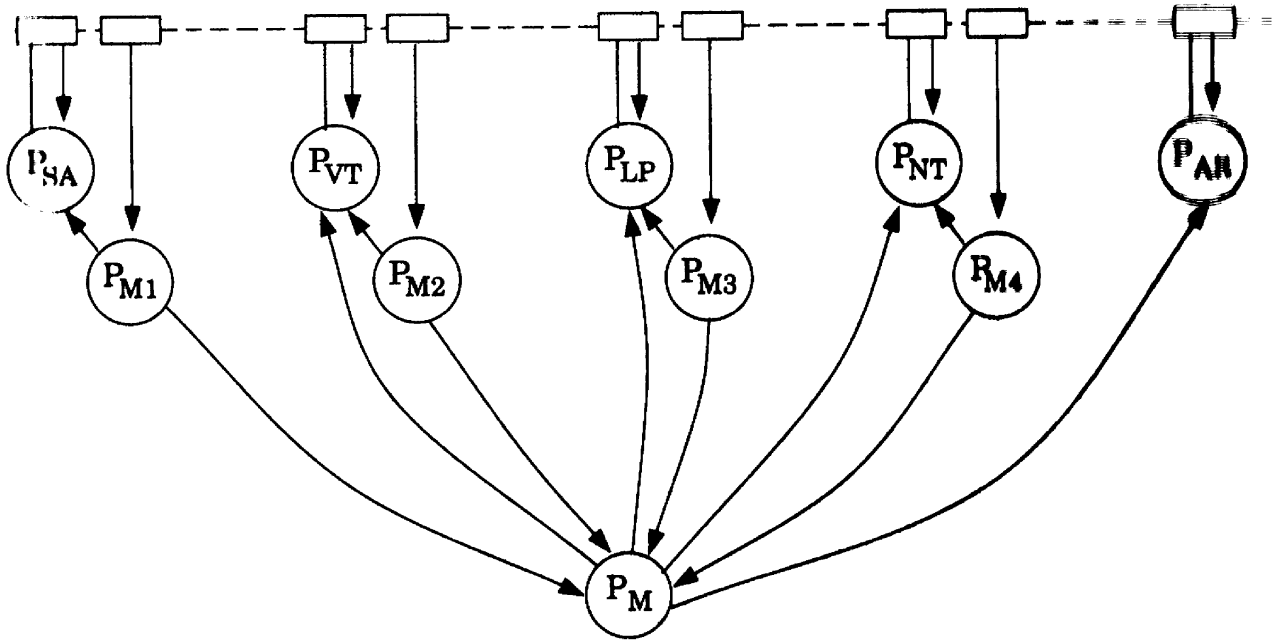


Here: linear chain of technical processes
often: parallel, concurrent processes!

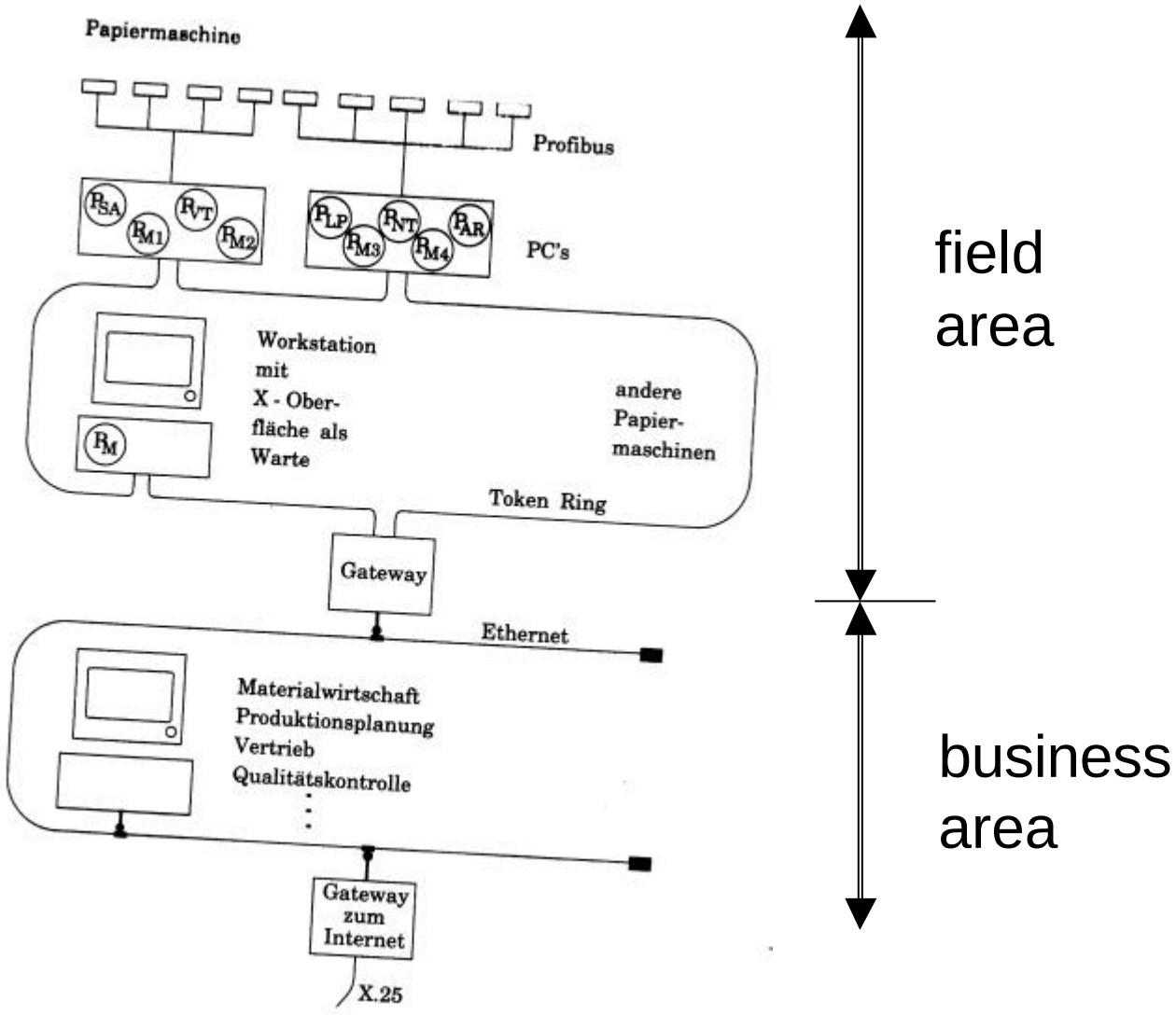
Example: paper producing machine, technical and computional processes



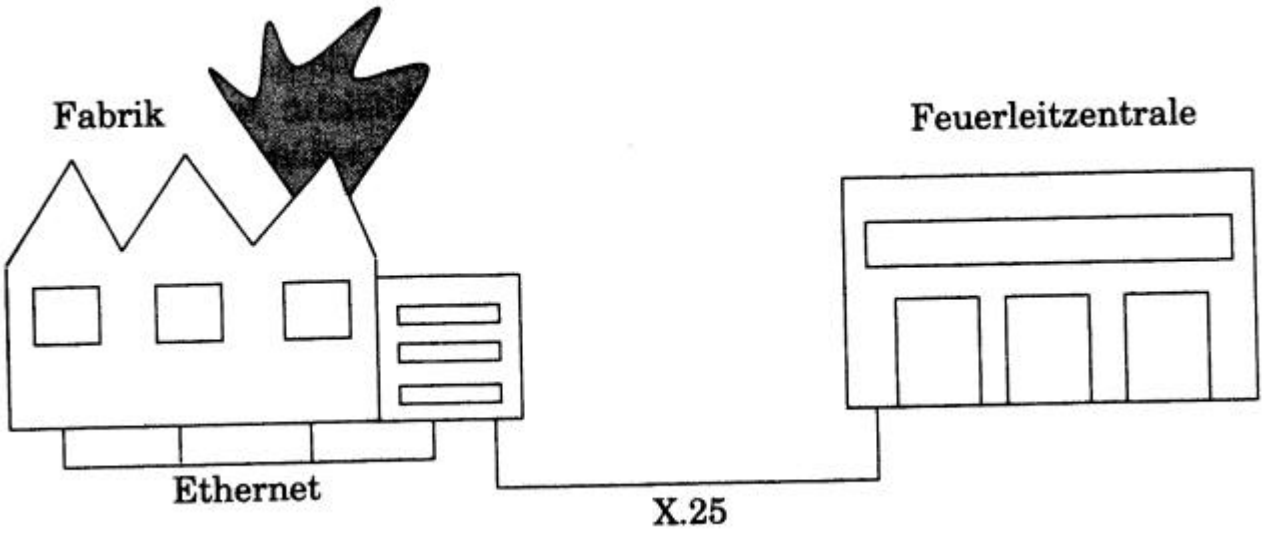
Example: paper producing machine, computional processes and measure process



Example: paper producing machines within a factory



Communication factory and fire watch



Scheme of a remote call