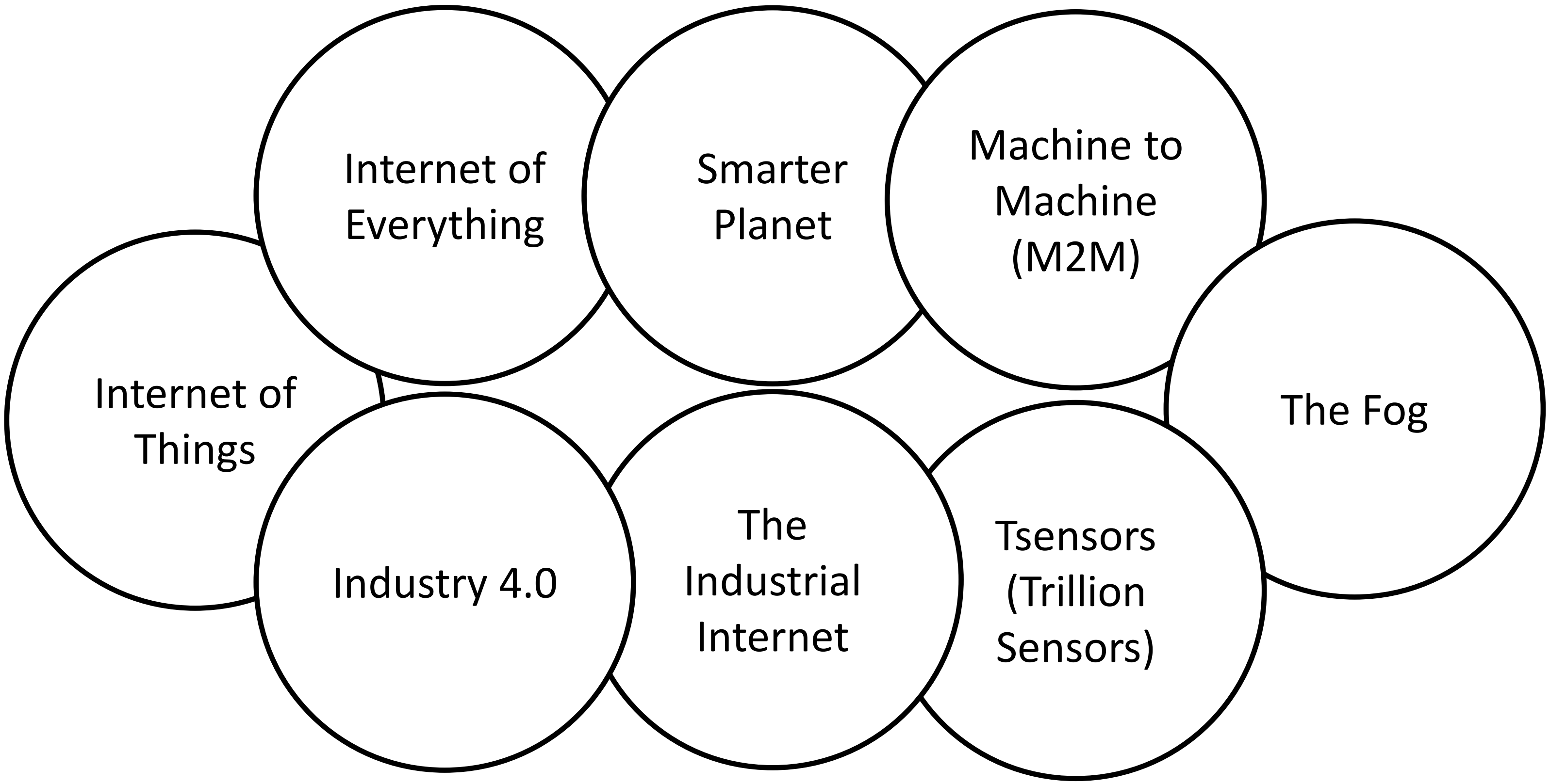
Embedded Information Systems

Introduction

September 15, 2020

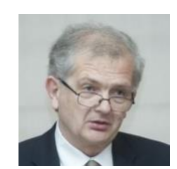
Smart Anything Everywhere

Many names– Similar meanings



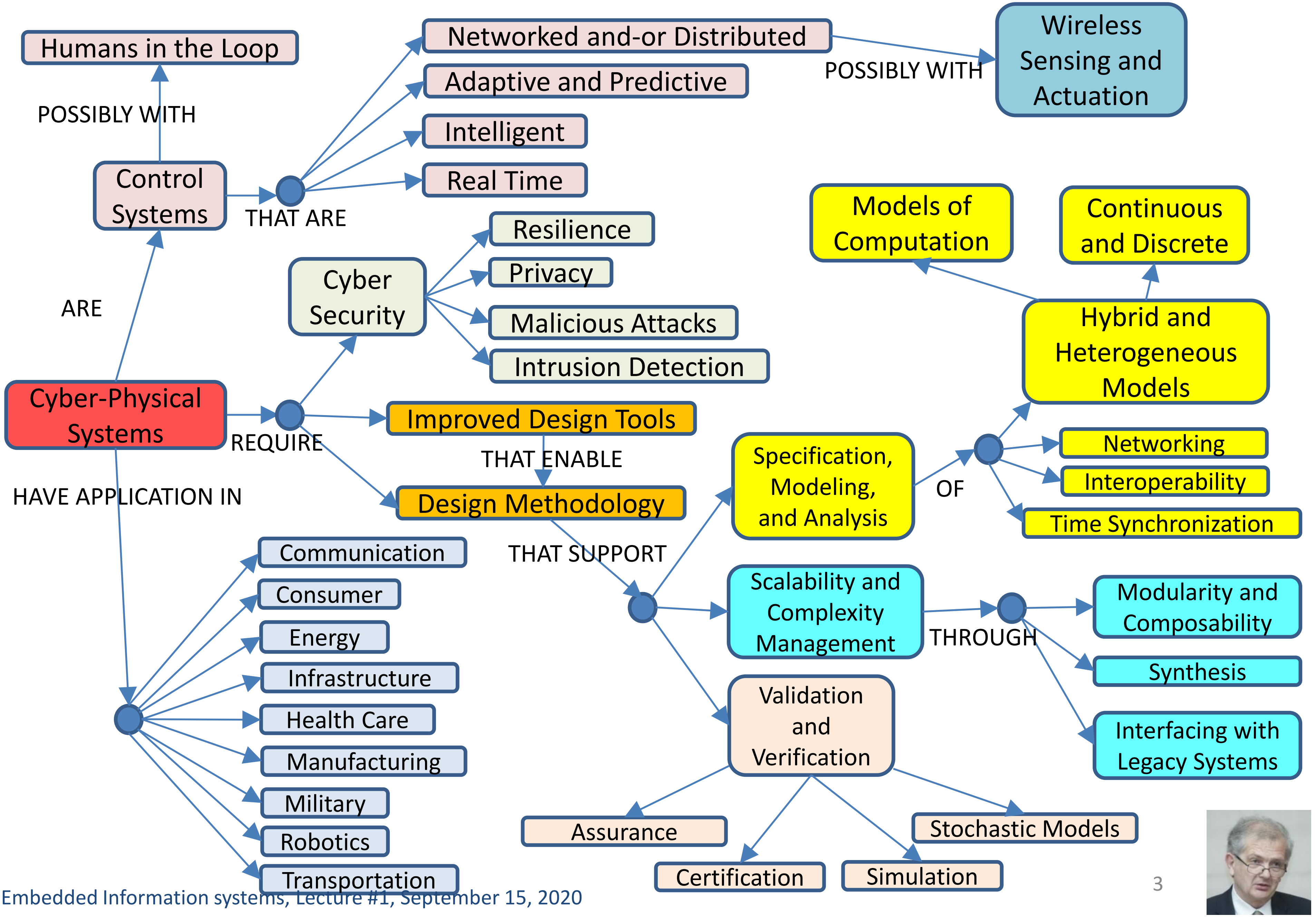
Cyber-Physical Systems

Networked embedded systems

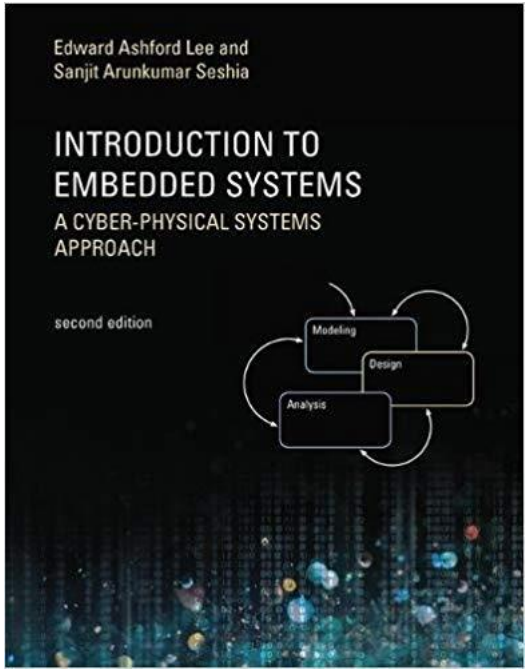


Cyber-Physical Systems – a Concept Map

http://CyberPhysicalSystems.org



A good reference:



Edward Ashford Lee, Sanjit Arunkumar Seshia

Department of Electrical Engineering and Computer Sciences (EECS) at UC Berkeley.

Introduction to Embedded Systems
A Cyber-Physical Systems Approach

Second Edition, LeeSeshia.org, 2017.

I Modeling Dynamic Behaviors

II Design of Embedded Systems

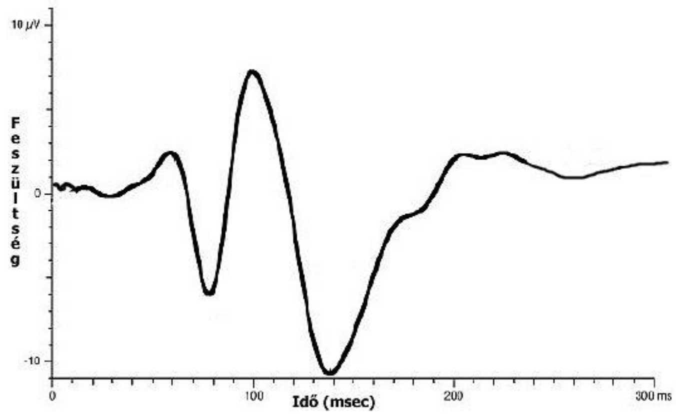
III Analysis and Verification

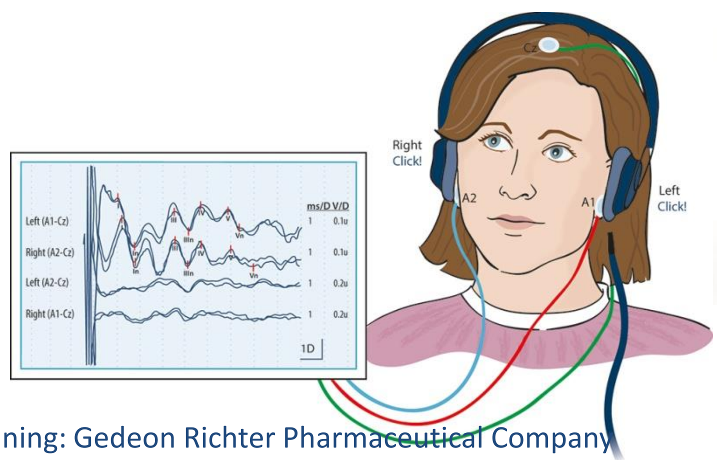
Recipient environments – embedded devices

EVOKED RESPONSES

– Responses of the central nervous system to external stimulations.

– These give information about the actual state of the nerve tracks, and about the processing of stimulations by the central nervous system.





Beginning: Gedeon Richter Pharmaceutical Company

Research Lab. Of Pharmacology ~1978

Cavinton … vasodilator drug …

PACEMAKERS

Treated diseases:

Parkinson-disease

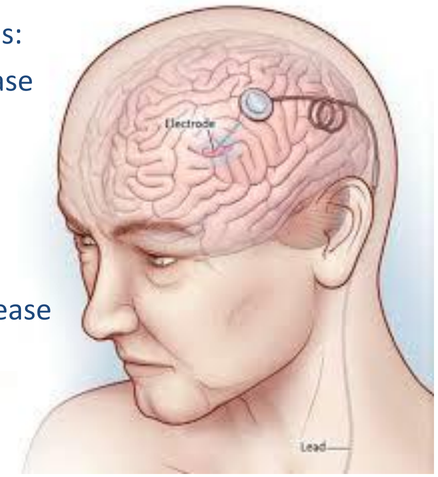
Anorexia

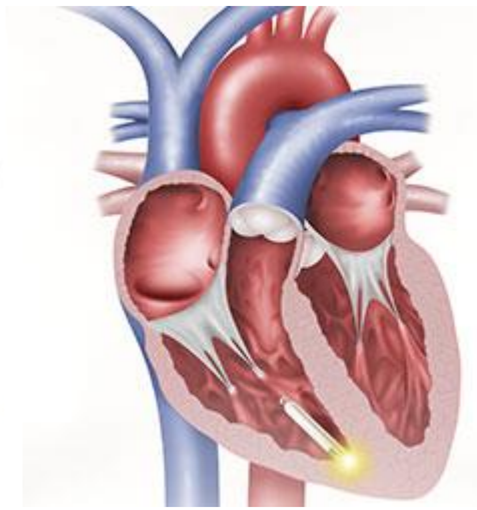
Epilepsy

Migraine

Depression

Altzheimer-disease

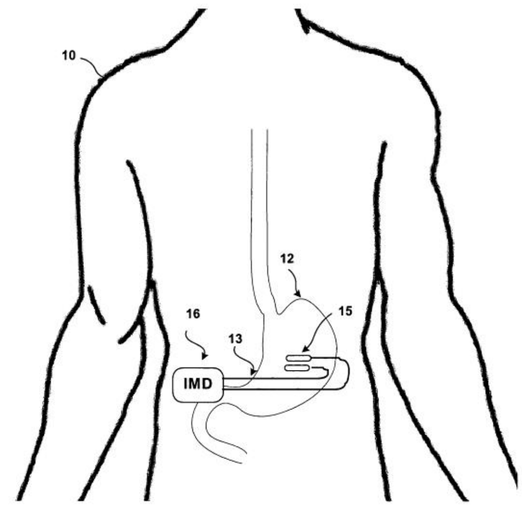




Treated diseases:

fibrillation

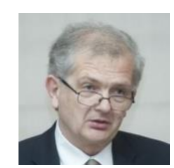
arrhythmia



Therapeutic goal:

to generate the feeling of satiety,

to avoid nausea

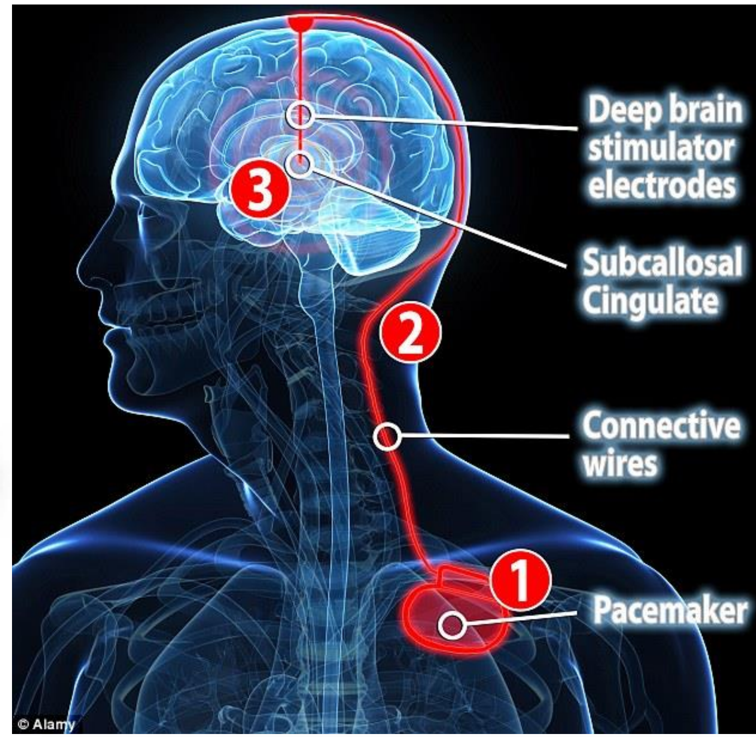


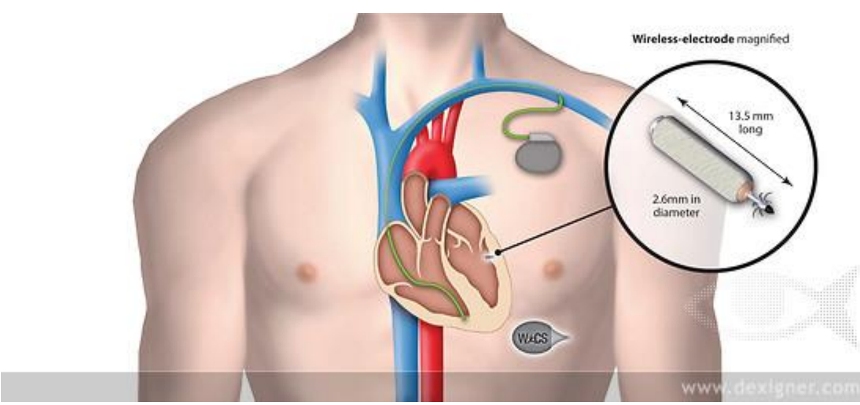
Received-embedded devices

Classical pacemaker
Since 2009 also with Internet connection
Implantation: 45’

Wireless pacemaker

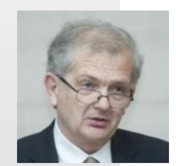
Implantation through catheter : 7’





13,5 mm long

2,6 mm in diameter



Embedded System Functions

Embedded system ~ Central nervous system

→ observes → analyses → decides → acts

The German automotive, automation and medical industry invests ~15 billion € for research and development of embedded systems pro year, while their annual income exceeds 500 billion €.

Main features:

Intensive information exchange
Autonomous operation
Dependability
„Invisibility”

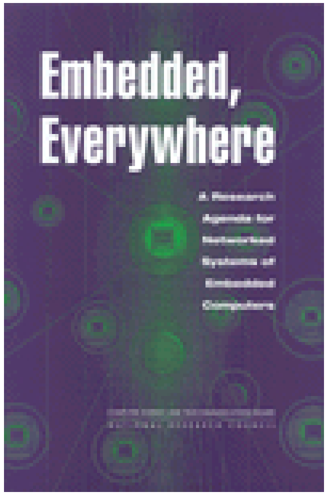
Alternative names:

Embedded System
Pervasive Computing
Ubiquitous Computing
Ambient intelligence

A possible definition:

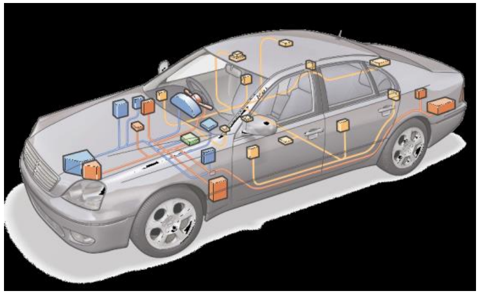
Embedded systems are **computer systems** which

- communicate intensively with their receiving physical/chemical/biological environment,
- operate autonomously,
- are highly reliable, and
- mostly “invisible”.
- Its elements have typically limited resources (memory, bandwidth, …),
- but at system level the resources prove to be ample



A Research Agenda
for Networked Systems
of Embedded Computers
National Academy of Sciences
(2001)

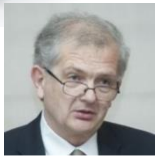
Fly-by-wire



Drive-by-wire

BMW 745i:
53 pcs. 8-bit,
11 pcs. 32-bit,
7 pcs. 16-bit processors,
2 000 000 line of code,
Windows CE OS,
Multiple network.

2% of the processors are used in IT and PC applications, 98% are embedded applications: vehicles, consumer electronics, mobile phones, etc.



The main actor is the embedded software

On one hand standardized hardware and software components (COTS) are applied, but the individual capabilities are provided by the software.

The components of the real systems interact more and more by computer mechanisms.

Within the premium category cars there are several thousand wires, and 70-100+ ECUs.

The embedded software is a universal system builder

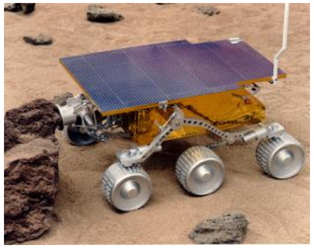
Consequences:

- On one hand the software absorbs its environment, while on the other it becomes part of the given application.
- The software meets both functional and physical requirements.

„... Software is Hard and Hardware is Soft ...”

Good news: using software many things are possible...
Bad news: using software many things are possible...

Challenges, lessons learnt:



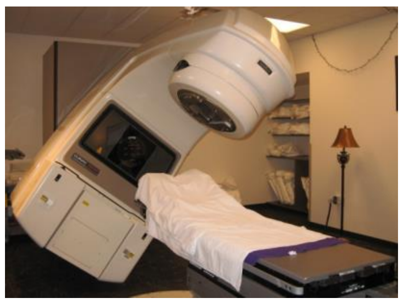
December 4, 1996. Mars Pathfinder mission. Priority inversion …

September 14, 1993. Warsaw Airport. Side wind, and sudden back wind + logical error. 2 dead, 54 injured …

February 25, 1991. Dahran: A Patriot missed a Scud missile. 28 dead, 97 injured. Software error, which was corrected already on 16 February, but was not delivered …

August 8, 1993. A fly-by-wire military aircraft crashed, because its reactions were too slow …

June 4, 1996: An Ariane 5. exploded due to software error: 64-bit floating-point -> 16-bit fixed-point …



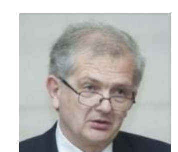
Between 1985 and 1987 the Therac-25 computer controlled radiation therapy system heavily overloaded 6 patients.

Between 1990 and 2000: 500 000 pacemakers were recalled!

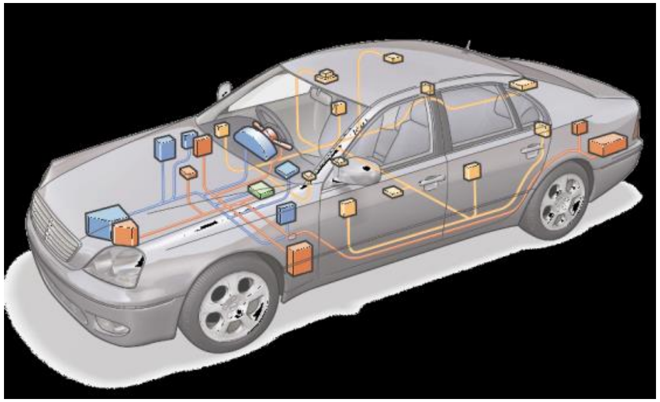
~75K Toyota Hybrid were recalled: “could enter a "fail-safe“ mode that shuts down the engine, allowing only limited operation using the electric motor. The problem, caused by a software error in the Electronic Control Module (ECM) system, triggers up to five warning lights while shutting down the engine.”

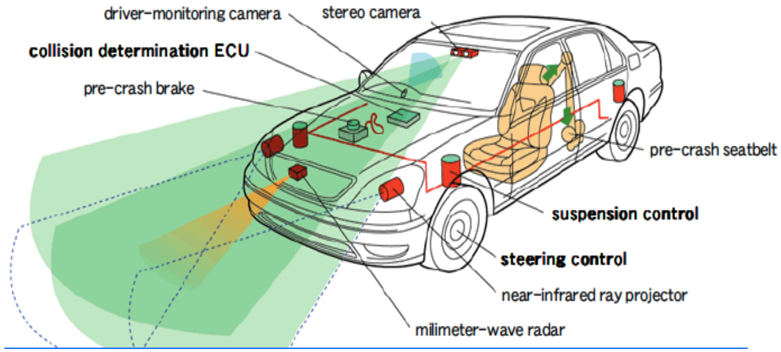
In the US **1.5M** Honda Accord, CR-V and Element were recalled: “to update the software that controls their automatic transmissions”

~8K Volvo S60 were recalled: to fix “ software for fuel pump units, as the software was not compatible with all fuel pumps and components.

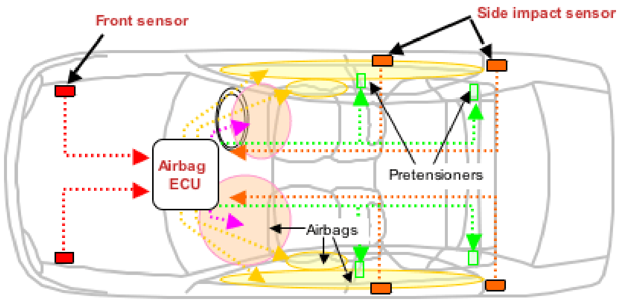


Cooperation of embedded components: systems of systems





Pre Collision Technology

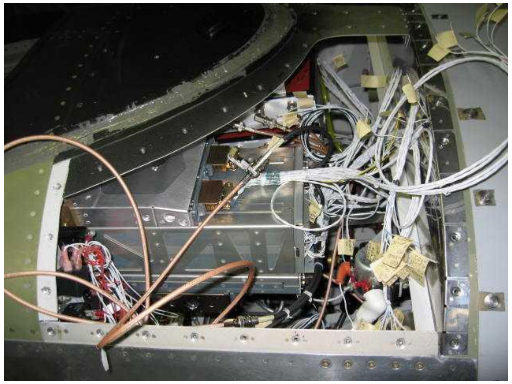


Air-bag system

Wiring harness is the 3rd most expensive car component after the engine and the body.
Wiring harness is the 3rd heaviest component after the body and the engine.
Its average weight is **100 kg**, its length ~**5km**.
Half of the cost of manufacturing the wiring harness is wage.

Several types of automotive networks:
CAN, LIN, Flexray, MOST, TTCAN, TT-Ethernet, …

Embedded devices and the internet



IEEE 802.3 based **Avionic Full-Duplex Switched Internet**: Applied in the Airbus A380, A400M and the Boing 787 Dreamliner!
IEEE 802.3bp standard announced in 2016: 1 Gbit/s **Internet in cars on single twisted pair of wire!** From 2019 on the market!
The internet connects **people, data, processes** and **things**. The capability of things to produce information is increasing!

Internet of Things:

It is a mapping of the physical world via internet to make it more knowable, followable and influenceable.

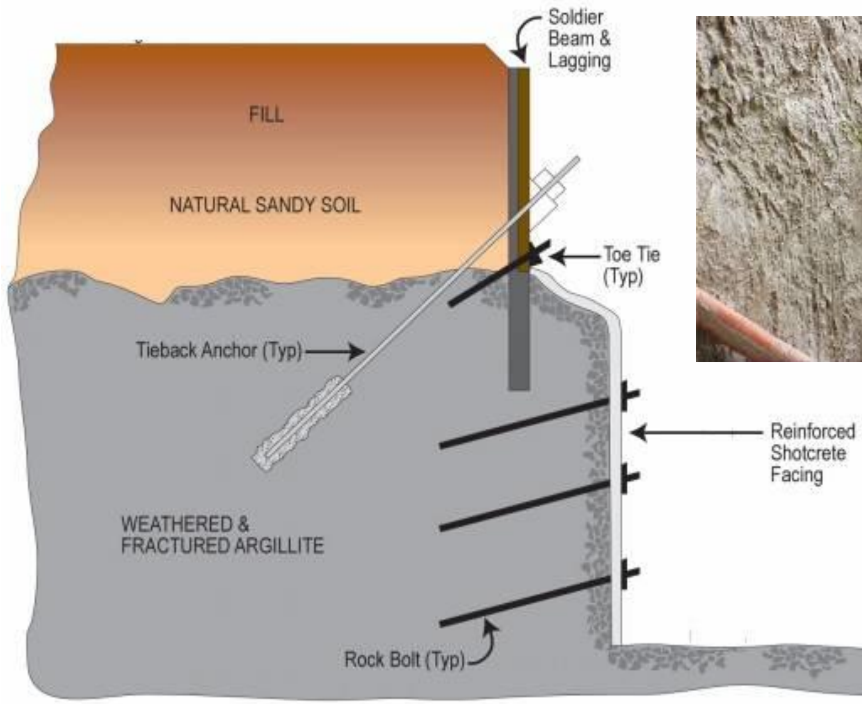
This results in the integration of embedded computers and their networks with physical processes.

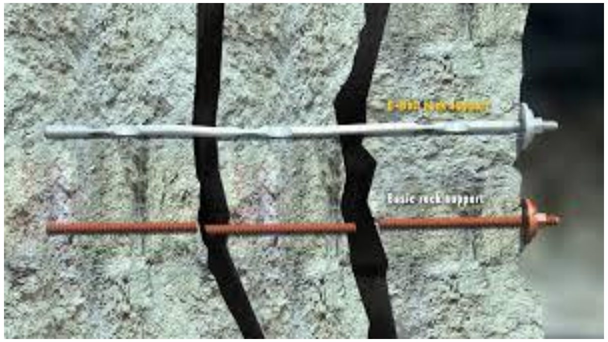
It includes such feedbacks, where physical processes influence calculations, and calculations physical processes.

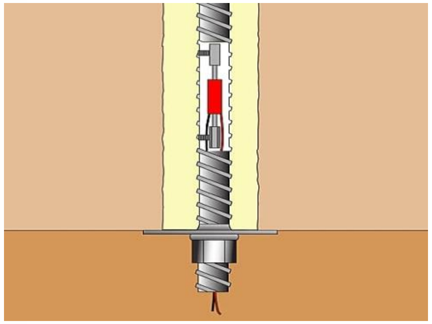
Recently the US Food and Drug Administration announced, that concerning cyber attacks more than 300 medical devices are unsafe: among them insulin pumps, pacemakers, infusion pumps, anesthetic devices.

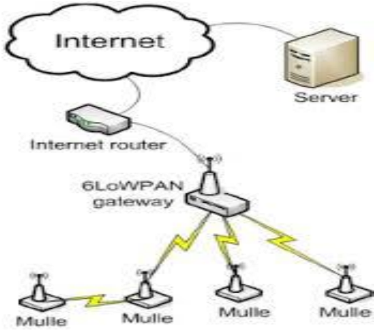
Systems of systems → Complexity → Security ?

Intelligent rock-bolt monitoring



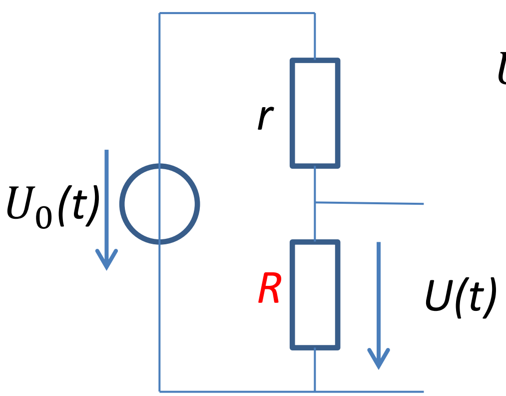






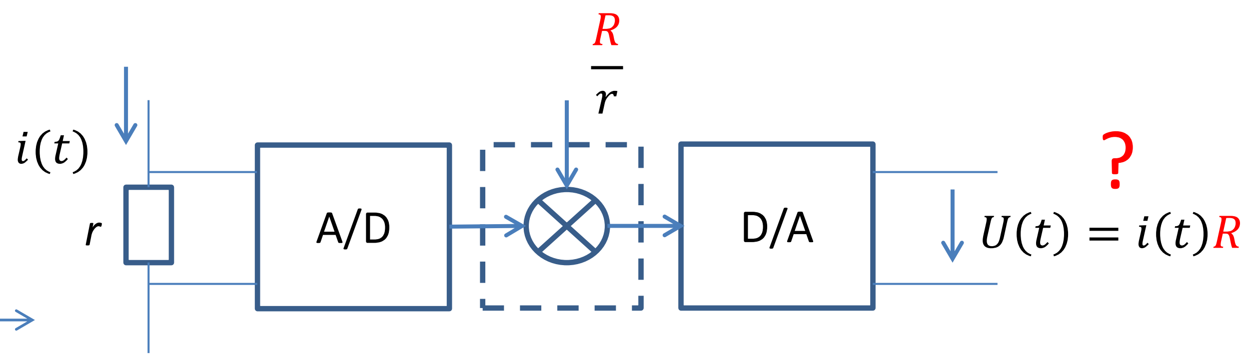
Modelling CPS systems

Example: Programmable voltage divider.



$$U(t) = U_0(t)\frac{R}{r+R} \qquad U(t) = i(t)R \qquad i(t) = \frac{U_0(t)}{r+R}$$

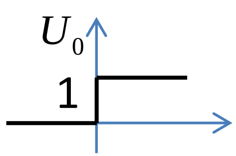
R can be tuned! Let's substitute R with the following circuit!



µP, DSP, "cloud", …

$$\boxed{U(t) = Ri(t-\Delta t)}$$

Consequence:



$$U(t=0) = 0 \qquad i(t=0) = \frac{U_0}{r}$$

$$U(t=\Delta t) = R\frac{U_0}{r} \qquad i(t=\Delta t) = \left(U_0 - R\frac{U_0}{r}\right)\frac{1}{r} = \left(1-\frac{R}{r}\right)\frac{U_0}{r}$$

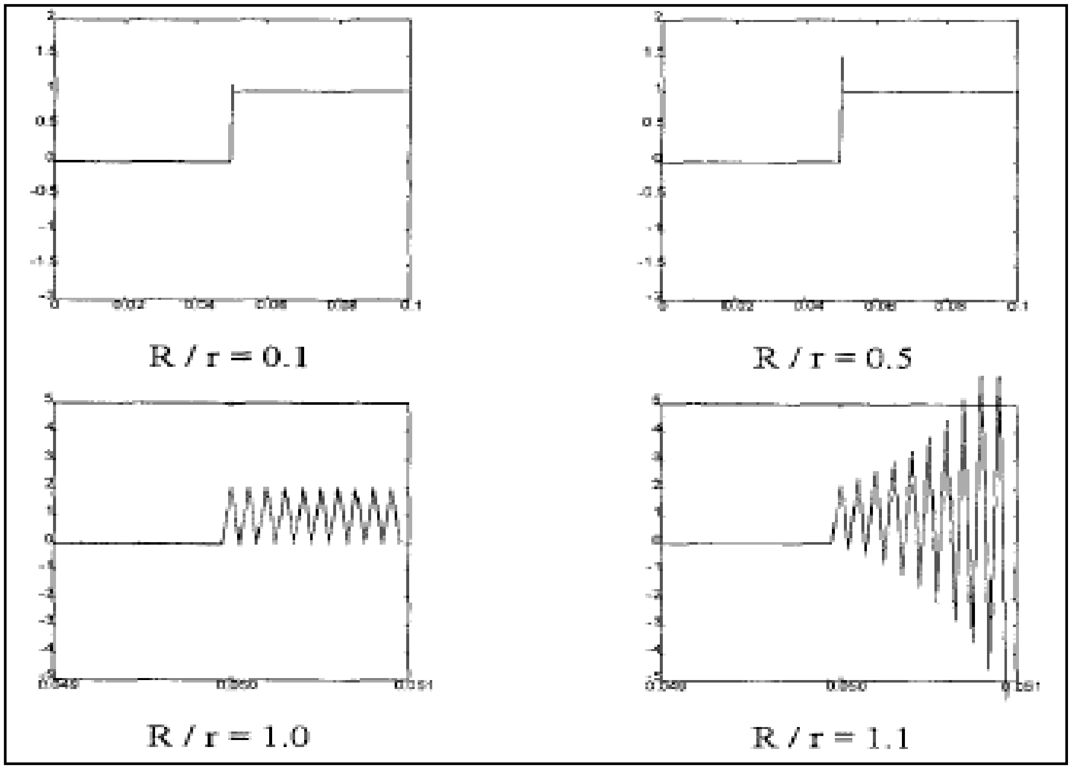
$$U(t=2\Delta t) = R\left(1-\frac{R}{r}\right)\frac{U_0}{r} \qquad i(t=2\Delta t) = \left[U_0 - R\left(1-\frac{R}{r}\right)\frac{U_0}{r}\right]\frac{1}{r} = \left(1-\frac{R}{r}+\left(\frac{R}{r}\right)^2\right)\frac{U_0}{r}$$

Modelling CPS systems

$$U(t=n\Delta t) = R\left(1-\frac{R}{r}+\left(\frac{R}{r}\right)^2 \mp \cdots \mp \left(\frac{R}{r}\right)^{n-1}\right)\frac{U_0}{r} \rightarrow U_0\frac{R}{r+R}$$

$$i(t=n\Delta t) = \left(1-\frac{R}{r}+\left(\frac{R}{r}\right)^2 \mp \cdots \pm \left(\frac{R}{r}\right)^{n}\right)\frac{U_0}{r} \rightarrow \frac{U_0}{r+R}$$

If $\frac{R}{r} < 1$



The delay can cause
overshoots,
damping oscillations,
constant magnitude oscillations,
growing magnitude oscillations
depending on the
parameter values.

The direct utilisation of continuous models is not always feasible!

Embedded systems of the future: trends and terms

Embedded Systems

- systems with embedded software…

Networked Embedded Systems

- communicating embedded systems…

Systems of Systems

- systems which communicate and cooperate …

Internet of Things and Services

- communication and cooperation of things and services …

Cyber-Physical Systems

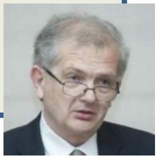
- integration of embedded systems and global networks to serve possibly all the users living on the globe

The purpose is to provide better quality:

In everybody's living, in medical services, in food production and distribution, in assisting elderly and handicapped people,

and to achieve these aims, to provide better quality:

In energy management, in transport, in environment protection, in disaster prevention, in life and property protection, …



European initiatives:

FP5, FP6, FP7 framework programs, Eureka ITEA, ARTEMIS: Advanced Research & Technology for Embedded Intelligent Systems, Horizon 2020, CHIST-ERA, Alliance for Internet of Things Innovation (AIOTI), Industry 4.0, ...

Major application fields:

- Efficient and secure mobility
- Well-being and health
- Sustainable production (food, energy, mining, …)
- Intelligent communities (intelligent and secure cities, spaces, …)

Key words of the challenges: safety critical systems, virtual reality, big data, systems of systems, cloud services, autonomous, adaptive and predictive control, internet of things, multi-core computations.

+ Horizon 2020: Leadership in enabling and industrial technologies

Smart Cyber-Physical Systems ICT-01-2014, ICT1.1-2016
Smart System Integration ICT-02-2014, ICT1.3-2016
Smart Anything Everywhere Initiative ICT1.4-2016
IoT and Platforms for Connected Smart Objects ICT-30-2015
R&I on IoT integration and platforms ICT7.3 – 2016
Smart Anything Everywhere Initiative ICT-04-2017

Challenges, research areas

In the field of data and signal processing:

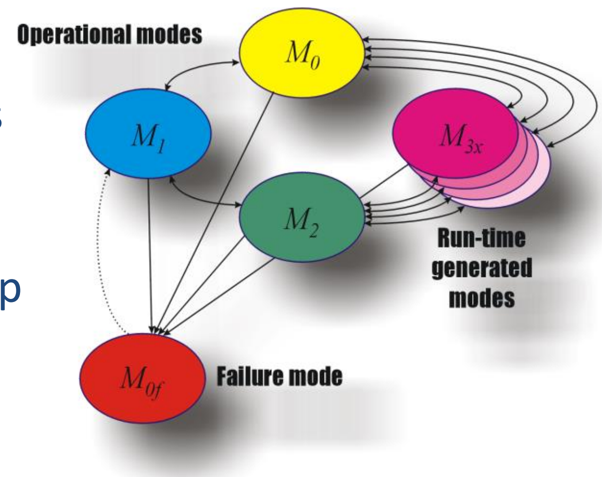
The quality of real-time data, and the related processing

- Accuracy, validity, loss of data
- Non-uniform sampling, synchronization of clocks and data
- Quantization errors in time and value
- Model-fitting, model-based and adaptive signal processing

In the field of system and control theory:

Control of hierarchical and distributed systems

- Stability of networked systems, passivity-based systems
- Adaptivity and cooperativity: reconfiguration, transient management
- Hybrid systems, hybrid simulation: hardware-in-the-loop
- Robustness, dependability, fault tolerance



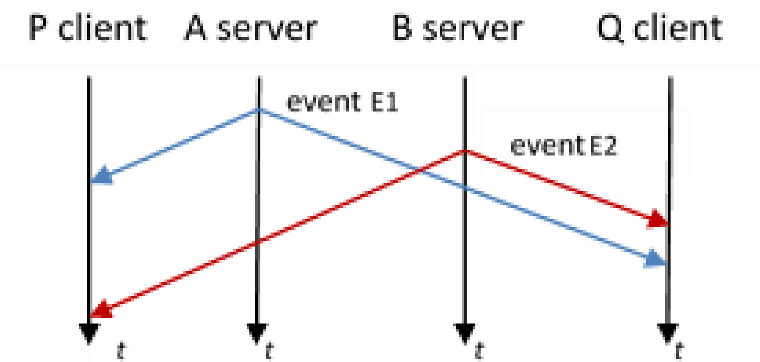
In the field of software development:

- Model-based system design
- Embedded virtualization, embedded systems using clouds

+ research in the field of verification, validation and certification of development tools, system and network software

Examples of specific time relations in embedded systems:

- **relativistic effect:** the time conditions of the communication through different channels may change the order of the event at the receiving node.



The figure illustrates, that in the case of client Q the message about event E2 precedes the message about event E1, which occurred earlier.

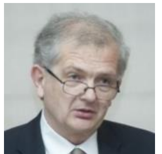
Such a situation might cause problems, if the decisions made at client Q depend on the order of the messages.

If the events E1 and E2 are not independent, after the arrival of the message about E2 to server Q, it might be reasonable to propose to wait for all those messages which were sent possibly at the same time instant or earlier as the message about E2.

This waiting time is called ***action delay***, which is the worst-case value of the possible message forwarding time for the case described above.

The necessary action delay can be calculated if the minimum and the maximum of the message forwarding time is known, i.e. for the message forwarding time the following is valid:

$$d_{min} \leq d \leq d_{max}$$



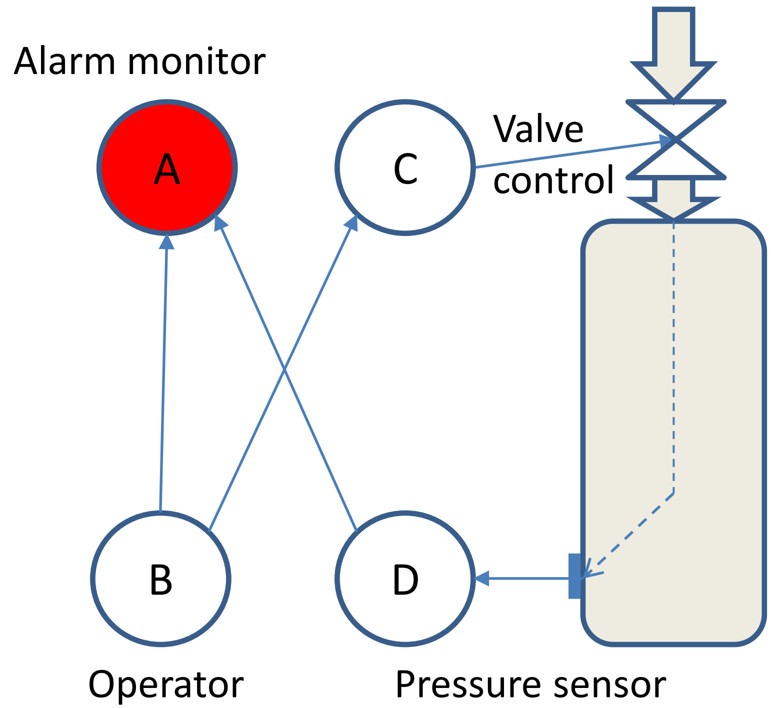
Quantities and variables in real-time systems

Example: We are monitoring the pressure within a container with a distributed system.

Node ***A***: alarm monitor,
Node ***B*** : operator,
Node ***C*** : valve control,
Node ***D*** : pressure sensor.
Possible messages:

$\mathbf{M}_{DA}$: indicates a drastic change of pressure,
$\mathbf{M}_{BC}$: operator command to change the valve,
$\mathbf{M}_{BA}$: It was an intentional change, no alarm.

Note: There is a hidden communication channel between the valve and the pressure sensor due to the operation of the physical system.



False alarm may occur, if
through $B \rightarrow C \rightarrow D \rightarrow A$ the information runs faster, than through $B \rightarrow A$.
To avoid this all actions of the alarm monitor should be delayed.
(Certain actions can not be withdrawn: a catapult, a shooting, etc.)

Note: the technological system itself implements a communication channel!

Quantities and variables in real-time systems

Action delay: we must wait until the **permanence** of the message.
Calculation of the delay:(1) If the global time is available:
$t_{permanent} = t_{sent} + d_{max} + 2g$, where d_{max} is the worst-case value of the message delay, and g stands for the resolution of the clock. (2) If the global time is not available:
$t_{permanent} = t_{sent} + 2d_{max} - d_{min} + g_l$, where d_{min} is minimum message delay, and g_l stands for the resolution of the local clock.

In the second case the delay is larger, because the sending time is not known, while in the first case it can be calculated from the time-stamp sent with the message.

The difference between the two cases is $d_{max} - d_{min}$ that can be large.
In the case e.g. of a token controlled bus if the token round takes 10 *ms*, while message forwarding time is always 1 *ms*, then d_{max} = 11*ms* and d_{min} = 1 *ms*, since if the token just left we must wait for 10 *ms*.

Comments:

- To understand the calculation of the action delay, imagine that you are an external observer, who knows the time of every event, and is familiar what is known and what is not at the different nodes.
- A RT image can be utilised only after reaching **permanence**. If this time exceeds the **time accuracy** of the image, only the state estimation can help.

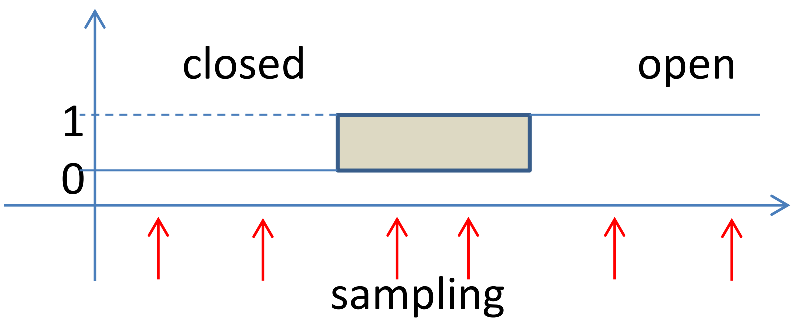
Quantities and variables in real-time systems

Permanence: it means, that the message/information becomes permanent, it will not be changed or modified. A message becomes permanent, if the receiving node knows that all the messages sent before the current message already arrived to the receiving node, or will never arrive.

- ***Real-time variables*** (RT entities): state variables, like e.g. fluid flow, setpoint of a controller, required position of a valve.

  They have static and time-dependent (dynamic) attributes.
- Every **RT variable** is within **the sphere of control** (SOC) of that subsystem, which is permitted to modify its value. Outside the SOC the **RT variable** is only readable.
- An RT variable can have either discrete or continuous value.
- A discrete **RT variable** can be undefined.

  Example: just opening garage gate: it is neither opened nor closed.



Quantities and variables in real-time systems

Observations: the value of the RT entity at given time instants

Observation =<name, observation time, value>

- *Observations in distributed systems:*

  if the global time is not available, then usability of the time stamps is limited. Many times the arrival time is used as observation time.
  This can cause considerable error in state estimation.

- *Indirect observations:*

  many times direct access to the quantity to be observed is not possible. In such cases model-based observations are made.
  (E.g. measurement of the internal temperature using observations on the surface).

- *State observation:*

  every observation produces such a value, which can be interpreted separately.
  It is quit typical that for these observation periodic sampling is applied.

- *Event observations:*

  the event is a change in the state at a given time instant.

  Since the observation itself is an event, therefore it is impossible the direct observation of the event within the controlled object, only its consequences can be observed.

Quantities and variables in real-time systems

Image of a real-time variable:
represent the RT variables within the computer program,
that are characterized by their accuracy in time and value, and even by their validity.

The RT image:
is an observation of the actual state, or an event.

RT objects:

An RT object within a node of the distributed system is a **container**, which contents an RT variable or its image.
To every object belongs a given accuracy **clock**.
Every clock tick activates an **object procedure**.
If this activation is periodic, then we are talking about **synchronous RT object**.
The **distributed RT objects** are present in different nodes as copies.
A possible example can be the **global clock**, which is present at different nodes as a **copy** operating with Π precision.

Accuracy in time:
The time of the information acquisition by the observation differs from the time of the utilisation within the computer. During this time difference the observed value changes.
The accuracy in time is defined as an interval $d_{accuracy}$ during which the change of the value is still tolerable concerning the specifications of the actual application.

Quantities and variables in real-time systems

Example: Some engine parameters are given in the Table below together with their magnitude accuracy and the corresponding time intervals.

RT image	max. change	accuracy	accuracy in time
Piston/cylinder position	6000 rpm	$0.1°$	$3\mu sec$
Gas pedal position	100%/sec	1%	10 msec
engine load	50%/sec	1%	20 msec
Oil and water temperature	10%/min	1%	6 sec

Among the accuracy intervals of the RT images the difference is more than 6 magnitude.

In the case of the piston position such an accuracy can be provided only with state estimation (prediction) within the program.