1.1 Definition

**Example: Pacemaker**
- one processor
- ca. 500,000 LoC (lines of code)

**1.1 Area’s Confine**
- Examples for use of real-time systems:
  - Different application domains
  - Mostly embedded systems

Legend:
- To get a better orientation, slides and handouts are marked with icons that indicate, what kind of knowledge is presented:
  - Concepts / models / definitions / discussions
  - Algorithms / implementations
  - Evaluations / analysis
  - Examples
Example: BMW 7 Series

- ca. 70 control units, in part with multiple processors
- ca. 100 million LoC

Example: Boeing 777

- ca. 1200 processors
- ca. 4 billion LoC

What is Real Time / a Real-Time System?

- There exist quit a lot, partly confusing beliefs, what are the properties of a real-time system
- Real-time systems have to...
  - be fast?
  - be predictable?
  - provide results in a timely manner?
  - be secure?
  - act error-free?
  - operate even in presence of a failure?
- Need for a definition

Definitions

DIN 44300: Real-Time Operation

Echtzeitbetrieb ist ein Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind derart, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind.

Approx. translation:

Real-time operation of a computer system is an operation mode, where programs that process data are always ready in the way, that allows to provide the results within a given period of time.

Wikipedia: Real-Time Computing

[...] real-time computing (RTC), or reactive computing, is the study of hardware and software systems that are subject to a "real-time constraint" – i.e., operational deadlines from event to system response. Real-time programs must execute within strict constraints on response time.
1.1 Definition

**Kopetz Real-time computer system**
A real-time computer system is a computer system, in which the correctness of the system behavior depends not only on the logical results of the computation, but also on the physical instant at which these results are produced.

**Krishna & Shin: Real-time system**
Any system where a timely response by the computer to an external stimuli is vital is a real-time system.

---

**Socratic Dialogue**

**Socrates:**
What means “timely”?

**Answer:**
It means that tasks in a real-time system have to obey a deadline.

---

**Socrates:**
Does “deadline” mean that any task really has to be finished by that time?

**Answer:**
Not necessarily. Sometimes yes: If landing of an airplane is controlled by a computer, missing a deadline may lead to a crash. Sometimes no: In a video game, nothing (really) bad happens in this case.

---

**Socrates:**
Does “deadline” mean that any task really has to be finished by that time?

**Answer:**
Not necessarily. Sometimes yes: In a nuclear reactor delayed reaction may lead to a disaster. Sometimes no: Getting a single measurement to late may not be a problem if the next measurement is on time.
1.1 Definition

Socratic Dialogue

Socrates: What means “finished”? Is there an exact criterion when a task is finished?

Answer:

Not necessarily.

Sometimes yes: For example, transferring money from one account to another may depend on calculating the sum of different accounts.

Sometimes no: If an application calculates the value of \( \pi \) it may decide to stop earlier and use a less accurate result.

Socrates: Again to the deadline: What happens if a task does not obey its deadline? Will it be terminated then?

Answer:

It differs.

If you are passenger of an airplane that crashes due to a missed deadline than this is not relevant both for you and the computer.

On the other hand, for a video conference it may be useful not to skip a delayed frame.

In general the value of a tasks result is decreasing for missed deadlines – in some cases slowly, in other cases abruptly.

Different Time-Value Functions

Socrates: Doesn’t this imply that any computer is a real-time system?

Answer:

Somehow yes. Taking the definition very precisely, any computer is a real-time system because nobody wants to wait forever for a result, e.g., in response to a keystroke.
1.1 Definition

Socratic Dialogue

**SOCRATES:**
If it includes all computer, isn’t the definition **senseless**?

**Answer:**
Okay, in order to stop this discussion, here is a new definition:

Refined Definition

A real-time system is any system that the lecturer sees as such. 😞

*In general, considerable work in real-time system design is spend on how timing constraints can be met.*

Misconceptions

“**Misconceptions about Real-Time Systems**”

- “There is no science in real-time-system design.”
  - We will that reconsider later...

- “Advances in supercomputing hardware will take care of real-time requirements.”
  - Argument of a salesman: “Buy a faster computer!”
  - But: In case of a deadlock, this is not an improvement...

*Meeting via [Sta88], access via https://www.tu-chemnitz.de/~werne/restricted-docs/Stankovic88Misconceptions.pdf*

Misconceptions (cont.)

- “Real-time computing is equivalent to fast computing.”
  - No. Time needed for computation has to be **predictable**.

- “Real-time programming is assembly coding and driver programming.”
  - This may be necessary in some cases but as always in computer science we want to automate as much as possible.
  - Real time needs **conceptual** understanding.

- “Real time is performance engineering.”
  - **Timeliness** is more important than speed.

- “Real-time problems have all been solved in other areas of computer science or or operations research.”
  - In computer science usually **peak performance** or **average performance** are of interest.
  - Operation research usually deals with **stochastical** models.

- “It is not meaningful to talk about guaranteeing real-time performance when things can fail.”
  - This is no reason **not to improve** the design.
  - A system is as weak as the weakest element. It is our goal to ensure that operating system or control software are **not** that weakest element.
What is the Problem?

Consider the following guarantee:

**PC Manufacturer warrants that**

(a) the SOFTWARE will **perform substantially in accordance** with the accompanying written materials for a period of ninety (90) days from the date of receipt, and

(b) any Microsoft hardware accompanying the SOFTWARE will be **free from defects** in materials and workmanship under normal use and service for a period of one (1) year from the date of receipt.

Hardware may fail, software is often faulty from beginning ⇒ software crisis

What is the Problem? (cont.)

Something is next to trivial, something is impossible

We are interested in the area in between

- Window of scarcity
- Example multimedia:

NEWTON'S Concept of Time

- Law of inertia is valid with respect to an (acceleration less) inertial systems
- Different inertial systems that are related to each other ⇒ there is one absolute time (and space)

**NEWTON**: Absolute dimension (beside space) where reality exists.

**EINSTEIN**: Relative dimension as part of space-time.

**Today**: Direction of entropy’s increase.
**EINSTEIN's Concept of Time (1905)**
- Time is ... 
  - neither independent nor constant
  - part of the space-time (Minkowski space)
- Constant here: speed of light
- Consequences:
  - There is no space-independent simultaneity
  - Speed of time depends on state of observer
- Example: Lorentz transformation

Moving at speed of $v$ in direction $x$:

$$t' = \frac{t - \frac{x}{c^2} v}{\sqrt{1 - \frac{v^2}{c^2}}} \quad x' = \frac{x - vt}{\sqrt{1 - \frac{v^2}{c^2}}} \quad y' = y \quad z' = z$$

**Thermodynamic Arrow of Time**
- Laws of nature usually do not differentiate between past and future
- Symmetry of charge, parity and time (CPT symmetry, PAULI, 1955)
- **However:** CP symmetry (charge and parity) can be broken
- Corresponds to Second Law of Thermodynamics (CLAUSIUS, 1854)
- Inversion of the Second Law to define time: direction of entropy's increase defines **thermodynamic arrow of time**

For practical reasons usually Newton's concept of time is still in use (e.g., in this lecture).

**Mathematization**
- In the most cases, Newton's concept of time is sufficient
- Newton's concept of time is usually mathematical reduced as follows:

Time is a set $M$ of discrete, distinguishable points in time with the relation "$<$" ("temporally before") that is defined as follows:

- **Trichotomy:** $\forall x, y \in M$ exactly one of the following conditions holds: $(x < y), (y < x), (x = y)$
- **Transitivity:** $\forall x, y, z \in M : (x < y) \land (y < z) \Rightarrow (x < z)$
- **Irreflexivity:** $\forall x \in M : \neg (x < x)$
- **Density:** $\forall x, y \in M : (x < y) \Rightarrow \exists z : (x < z < y)$

- Trichotomy, transitivity, and irreflexivity imply that "$<$" is a strong total order

**Measuring Time**
- Measuring of time is a problem since time cannot be repeated
- Issue of standardization
- First attempt: astrological phenomena
  - Day: Duration between two sun zeniths (solar day)
  - Year: Duration that the Sun takes to return to the same position in the cycle of seasons, e.g., for example, the time from vernal equinox to vernal equinox (tropical year)
  - Month: period of the Moon's phases
### Subdivisions

- Naive division of day (and night) into 12 hours; fixed phenomena: sun raise and sun set
  - In use until 15th century
  - **Issue**: length of hours differs; except **equinoxes**
- Compensation by introduction of the **mean solar day**
- **Second**: $\frac{1}{86400}$ of the mean solar day

### SI Second

- Astronomic phenomenons are not "precise enough"
- **Remedy**: definition without help of astronomy

**Second due to SI (1967)**

A second is the duration of 9,192,631,770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the caesium-133 atom.

### 1.3 Real-Time Entity

**Real-Time Entity**

- Embedded system: overall system consist of computer and non-computing components
- The state of the non-computing components (and of the environment) are of interest
  - **state variables**
    - **Examples**: position of robot manipulator, speed, ...
- Relevant state variables are called **real-time entities**
  - **Relevance** due to two reasons:
    - The application want to manipulate it
    - It manipulates/has impact to the application
- Statements are valid for limited time only

**Real-Time Image**

- The real-time entity is mapped to a state of the computation system
  - Mapping with help of sensors; sometimes multiple transformations
- The mapping result is called **real-time image**
  - The applications uses always the image, but tries to impact the real-time entity
  - Mapping results may differ from real-time entity in value and time
  - Difference in time: temporal accuracy interval
- The goodness of the real-time image is an important design parameter
1.3 Real-Time Entity

Accuracy

Sphere of Control

- Each real-time entity may exist within a sphere of control (SoC)
  - Within a SoC, a real-time entity can be influenced
    - Influence works indirectly (via actors)
  - Out of a SoC, a real-time entity may be observed, but not controlled

Example car:
- Real-time entities: traffic light, other cars, own speed
- Traffic lights, other cars, their speeds, etc. can be observed
- Own speed can be controlled

References


