Software Engineering for Software-Intensive Systems: Introduction

Assistant Professor Dr. Holger Giese
Software Engineering Group
Room E 3.165
Tel. 60-3321
Email: hg@upb.de
Outline

I Introduction
II Foundations
III The Development Life Cycle
IV Requirements
V Analysis & Design
VI Implementation
VII Verification & Validation
VIII Summary and Outlook
I Introduction

I.1 Definitions & Terminology

I.2 Examples
I.3 System Characteristics
I.4 Application Domains
I.5 Resulting Challenges & Future Challenges
I.6 Discussion & Summary
I.7 Bibliography
I.1 Definitions & Terminology

A software-intensive system is any system where software contributes essential influences to the design, construction, deployment, and evolution of the system as a whole.

[IEEE-Std-1471-2000]

Observation ([EU-NSF-SIS2004]):

Software has become a key feature of a rapidly growing range of products and services from all sectors of economic activity. Software-intensive systems include

- large-scale heterogeneous systems,
- embedded systems for automotive applications,
- telecommunications,
- wireless ad hoc systems,
- business applications with an emphasis on web services etc.

Our daily lives depend on complex software-intensive systems, from banking to communications to transportation to medicine.
1.1.1 System Types

Example system types (can overlap):

- **business management information systems**
  (e.g., resource planning)
- **net-reliant systems**
  (e.g., command and control)
- **infrastructure systems**
  (e.g., net-centric enterprise services)
- **embedded systems**
  (e.g., control applications)

Possible Overlapping

- Business system
- Embedded system
- Net-reliant system
- Infrastructure system
Business Systems

Business systems acquisitions typically consist of:

- a small set of large commercial off-the-shelf (COTS) products (e.g., information system components)
- adapted/configured/ to support the business and management functions of an organization
- Typical business systems include
  - financial management,
  - personnel management, and
  - enterprise resource planning.
Net-Reliant Systems

- Net-reliant systems provide functions that rely on data exchanges with physically disparate elements.
- These systems involve large amounts of data push (control) or data pull (awareness) function
- assist humans in awareness and decision-making processes
Network Infrastructure

- A network infrastructure system provides the equipment and capabilities necessary for the successful operation of net-reliant systems.
  - Complex: large database management and glue logic to execute and retrieve services across a Wide Area Network of varying security.
  - Simple: Local Area Networks

- Timeliness and robustness of services
- The network infrastructure is usually critical
Embedded Systems (1/2)

- Software embedded into a technical system that interacts with the real physical world, controlling some specific hardware.
- Interaction with the technical system via actors and sensors.
Embedded Systems (2/2)

Examples of embedded systems:
- avionics, such as inertial guidance systems, flight control hardware/software and other integrated systems in aircraft and missiles
- engine controllers and antilock brake controllers for automobiles
- home automation products, like thermostats, air conditioners, sprinklers, and security monitoring systems
- household appliances, including microwave ovens, washing machines, television sets, DVD players/recorders
- medical equipment
- measurement equipment such as digital storage oscilloscopes, logic analyzers, and spectrum analyzers
- programmable logic controllers (PLCs) for industrial automation and monitoring

**IMPORTANT:** Often the term embedded systems is also used for a much broader category of systems (nearly synonymously with software-intensive systems)
Importance of SIS/Embedded Systems


The importance of embedded systems is undisputed. Their market size is about 100 times the desktop market. Hardly any new product reaches the market without embedded systems any more. The number of embedded systems in a product ranges from one to tens in consumer products and to hundreds in large professional systems. [...] This will grow at least one order of magnitude in this decade. [...] The strong increasing penetration of embedded systems in products and services creates huge opportunities for all kinds of enterprises and institutions. At the same time, the fast pace of penetration poses an immense threat for most of them. It concerns enterprises and institutions in such diverse areas as agriculture, health care, environment, road construction, security, mechanics, shipbuilding, medical appliances, language products, consumer electronics, etc.

(cited from [Bouyssounouse&Sifakis2005, p. 2]
I.1.2 Involved Disciplines

- System Engineering
- Control Engineering
- Software Engineering
- Other Relevant Disciplines

⇒ The need for Integration
System Engineering

- Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem:
  - Operations,
  - Cost & Schedule,
  - Performance,
  - Training & Support,
  - Test,
  - Disposal,
  - Manufacturing

Relation to Software Engineering:
- Systems Engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs.
Control Engineering

- Control engineering is the engineering discipline that focuses on the mathematical modeling systems of a diverse nature, analyzing their dynamic behavior, and using control theory to make a controller that will cause the systems to behave in a desired manner.

- Control engineering has diversified applications that include electrical engineering, mechanical engineering, process control, science, finance management, and even human behavior.

Relation to Software Engineering:
- Control is challenging since it takes strong foundations in engineering and mathematics, uses extensively computer software and hardware and requires the ability to address and solve new problems in a variety of disciplines, ranging from aeronautical to electrical and chemical engineering, to chemistry, biology and economics.
Software Engineering

- The Establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.
  
  F.L. Bauer in [Buxton & Randell 1969]

- software engineering. (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).
  
  [IEEE-Std-610.12-1990]

Looking Beyond Software?

- For computer scientists, being told that creating software-intensive systems is difficult is not news. Even attendees at the 1968 NATO Software Engineering Conference in Garmisch, Germany, recognized the need to build, for example, a foundation for the systematic creation of software-intensive systems [3].

  (See [Freeman & Hart 2004])
Other Relevant Disciplines

- **Mechatronics** is the synergistic combination of
  - mechanical engineering ("mecha" for mechanisms),
  - electronic engineering ("tronics" for electronics), and
  - software engineering.
  - Purpose:
    - study of automata from an engineering perspective
    - controlling advanced hybrid-systems

- **Telematic** is the [integrated] use of telecommunications and informatics.
  - It is the science of sending, receiving and storing information via telecommunication devices.
  - Often used for Global Positioning System technology integrated with computers and mobile communications technology or for such systems within road vehicles (vehicle telematics)
  - **Vehicle telematic** systems may be used for a number of purposes, including
    - collecting road tolls and managing road usage,
    - tracking fleet vehicle locations and recovering stolen vehicles,
    - providing automatic collision notification,
    - providing location-driven driver information services,
    - remote diagnostics

[en.wikipedia.org/wiki/Mechatronic]

[en.wikipedia.org/wiki/Telematic]
The Need for Integration

The past (in most large projects):

- **System Engineering**: A set of interdisciplinary activities leading to a system that works and will meet stakeholder needs.
- **Software Engineering**: A set of processes that turn software requirements into working code.
- **Control Engineering** simply use extensively computer software and hardware.

The needed future:

- Software Engineering participate in establishing system requirements, in exploring design alternatives, and in integration and system test.
- Software Engineering must be integrated with Control Engineering in order to control future generation of systems of systems.
- Systems, Control, and Software Engineering must seek to integrate, not differentiate.
I Introduction

I.1 Definitions & Terminology

I.2 Examples

I.3 System Characteristics

I.4 Application Domains

I.5 Resulting Challenges & Future Challenges

I.6 Discussion & Summary

I.7 Bibliography
I.2.1 Example: VW Phaeton

Characteristics of the VW Phaeton:

- 61 electronic control units (ECUs); 45 different ones
- three coupled bus systems including one optical bus (3860 m cables; 64 kg)
- about 2500 signals in 250 CAN-messages
- 50 MB memory

⇒ Problem: complexity

http://www.heise.de/ct/03/14/170/
Functions (realized with Software)
Challenges

- Dramatic increase in complexity
- Supply chains with vendors and suppliers
- Heterogeneity in hardware and software
I.2.2 Example: The Denver International Airport Baggage System

The Denver International Airport has a modern, automated baggage-handling system designed by BAE Automated Systems, Inc. (In June, 2003 G & T Conveyor Company, Inc. acquired BAE)

See:
Automated Baggage-Handling System (1/4)

- The baggage handling system at an airport plays a crucial role. It makes the difference in an airport's ability to attract or keep a major airline **hub** ("an airport that serves as a central connecting point")

- A baggage-handling system has three main jobs:
  - Move bags from the check-in area to the departure gate
  - Move bags from one gate to another during transfers
  - Move bags from the arrival gate to the baggage-claim area

- **Conveyors** equipped with **junctions** and **sorting machines** automatically route the bags to the gate.
Automated Baggage-Handling System (2/4)

- Destination-coded vehicles (DCVs), unmanned carts (also named telecars)
  - propelled by linear induction motors mounted to the tracks
  - can load bags (without stopping)
  - can unload bags (without stopping)

a DCV (telecar)

load DCV

tunel system

unload DCV
Automated Baggage-Handling System (3/4)

DIA has:
- More than 5 miles (8 km) of conveyor belts
- More than 19 miles (30 km) of DCV tracks
- About 4,000 DCVs

⇒ enabling it to handle over 1,000 bags per minute.
Automated Baggage-Handling System (4/4)

Computer Equipment:
- 300 486-class computers distributed in eight control rooms,
- a Raima Corp. database running on a Netframes Systems fault tolerant NF250 server,
- a high speed optic network,
- 14 million feet of wiring,
- 56 laser arrays,
- 400 frequency readers,
- 10000 monitors, and
- 92 PLCs to control motors and track switches.

... and the development was a disaster (see SQA lecture)
Challenges

- Large scale project
- Embedded + Logistics
- Late integration of the Automated Baggage-Handling System into the overall system (system of systems)
- Long testing phase required!
I.2.3 Example: The Railcab Project

A system of autonomous shuttles that builds convoys to optimize their energy consumption (Railcab Paderborn):

- System of systems
- Hard real-time
- Safety-critical
- Self-Optimization (SFB 614)
Credits & Copyright

- AK Demonstrator meeting 26 Sep 02:
  - Slides from Michael Walther
  - Slides from Clemens Ettingshausen

- SFB Colloquium 2. and 3. Dez 02:
  - Slides from D1 (Markus Henke)
  - Slides from D2 (Clemens Ettingshausen)

- Media material etc.


- Current draft documents of the Railcab architecture
(A) New Logistic Concept

Difference:
- Passive tracks
- Autonomously routed shuttles

Shuttle:
- automatic and unmanned
- decentralized routing decision
- Information basis
  - Positioning system
  - mobile communication (between the shuttles and the track system)
  - On-board information system

Elements of the track system:
- Switchen, stations, railway crossings, shuttle deposits, etc.
- Connected via mobile communication
Customer View

- More comfort for passenger transport
  - flexible online booking
  - on demand transport
  - No change of the shuttle/train
  - active changes of the transport destination
  - On-board information system
  - customer specific rates (slow train, economic, express, individual)

- Better service for cargo transportation
  - flexible online booking (integration)
  - Delivery just-in-time
  - Specific transportation rates (std., express, kurier)
Comparison: Energy Consumption

10 Personen-Fahrzeug

Primär-Energie-Verbrauch in Wh pro Person / km

Anzahl NBP-Fahrzeuge im Konvoi

Gesamt-Personen-Zahl im Konvoi/Zug

- RailCab
- ICE
- Transrapid

Holger Giese
### Annahmen:
- Auslastung der Fahrzeuge je 100 %
- * Verbrauchswerte für jew. Betriebsgeschw. berechnet
- 100km-Strecke mit 1 Beschleun.-/abbremsvorgang

Verbrauchswerte für Fahrzeuge im Konvoi:
- Ohne Einfluß der Frontfläche, nur Verluste durch Roll-/ Lagerreibung

<table>
<thead>
<tr>
<th></th>
<th>Prototyp</th>
<th>NBP 10 Personen</th>
<th>NBP 16 Personen</th>
<th>ICE 3</th>
<th>Transrapid</th>
<th>DC Sprinter (Diesel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>max. Personenzahl</td>
<td>-</td>
<td>10</td>
<td>16</td>
<td>415</td>
<td>446</td>
<td>9</td>
</tr>
<tr>
<td>Gesamtbreite [m]</td>
<td>2,42</td>
<td>2,42</td>
<td>3,07</td>
<td>3,0</td>
<td>3,0</td>
<td>1,93</td>
</tr>
<tr>
<td>Gesamthöhe [m]</td>
<td>2,86</td>
<td>2,86</td>
<td>3,84</td>
<td>4,20</td>
<td>2,35</td>
<td></td>
</tr>
<tr>
<td>Gesamtlänge [m]</td>
<td>6,60</td>
<td>7,60</td>
<td>200,00</td>
<td>109,00</td>
<td>5,64</td>
<td></td>
</tr>
<tr>
<td>Leermasse [kg]</td>
<td>3,725</td>
<td>5,210</td>
<td>409,00</td>
<td>282,00</td>
<td>1,917</td>
<td></td>
</tr>
<tr>
<td>max. Nutzlast [kg]</td>
<td>1,000</td>
<td>1,600</td>
<td>33,00</td>
<td>35,50</td>
<td>1,15</td>
<td></td>
</tr>
<tr>
<td>zul. Gesamtgewicht [kg]</td>
<td>4,725</td>
<td>6,810</td>
<td>442,00</td>
<td>317,50</td>
<td>3,500</td>
<td></td>
</tr>
<tr>
<td>Rotor-/Statorbreite [m]</td>
<td>0,60</td>
<td>0,60</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rotorlänge [m]</td>
<td>3,33</td>
<td>4,76</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gesamtleistung [kW]</td>
<td>111</td>
<td>125</td>
<td>8,000</td>
<td>15,000</td>
<td>115</td>
<td></td>
</tr>
<tr>
<td>Betriebs-Geschwindigkeit [km/h]</td>
<td>160</td>
<td>160</td>
<td>300</td>
<td>430</td>
<td>160</td>
<td></td>
</tr>
<tr>
<td>Abg. Leistung f. Konstantfahrt bei Betriebs-Geschw. [kW]</td>
<td>85</td>
<td>88</td>
<td>6,010</td>
<td>13,087</td>
<td>115</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Einzelfahrt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gesamtwirkungsgrad ab Primärquelle</td>
<td>-</td>
<td>0,242</td>
<td>0,242</td>
<td>0,334</td>
<td>0,349</td>
<td>0,340</td>
</tr>
<tr>
<td>Primär-Verbrauch pro Sitzplatz / km bei 160 km/h [Wh]</td>
<td>219</td>
<td>142</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>235</td>
</tr>
<tr>
<td>Primär-Verbrauch pro Sitzplatz / km bei 100 km/h [Wh]</td>
<td>59</td>
<td>39</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>57</td>
</tr>
<tr>
<td>Wirkungsgrad Motor</td>
<td>-</td>
<td>0,625</td>
<td>0,625</td>
<td>0,844</td>
<td>0,873</td>
<td>0,375</td>
</tr>
<tr>
<td>Konvoifahrt</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prim.-Verbr./Sitz/km im Konvoi bei Betriebsgeschwindigkeit [Wh]</td>
<td>3 Shuttles/ 83</td>
<td>3 Shuttles/ 57</td>
<td>Komplettzug/ 106 *</td>
<td>Komplettzug/ 121 *</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>
Economic Efficiency

Instead of long term planned schedules for trains:

- flexible online management of the transport requests
  - locale assignment to shuttles
  - extra shuttle transfers for automatic load balancing
  - traffic jams are avoided
  - Takes customer specific requirements into account (costs vs. time, …)

- Cargo and passenger transport

- Minimize the operational costs
  - Shuttles without job drive to a local deposit
  - Automatic building of convois saves energy
  - small, flexible units and almost really required rides

⇒ better system utilization & cost minimization
(B) Prototype HW (Scale 1:2,5)

- Contains all major components of the final shuttle.

Technical data:
- Length 3 m
- Width 1,2 m
- Height 1,2 m
- gage of the track 600 mm
- Weight ca. 1000 kg
- Speed max. 36 km/h

The prototype has a number of additional sensors and data processing components for testing purposes.
Internal Structure
Prototype (Scale 1:2.5)
Structure of the Undercarriage

Two identical Undercarriages per shuttle (modular construction)
Traction Module & Brake
Protection
Emergency Brake

- Emergency brake for safety reasons
- Parking
Guidance Module

Normal:
- Damping the guidance
Here:
- Active guidance
Guidance Module & Linear Drive

Construction:
- Decoupled wheels
  - Minimize abrasion
  - Active guidance needed!

Active guidance:
- Avoid contact between the wheel flange and the head of the rail
- Artificial sinus curve
- Preview can be used to optimize guidance
Passive Switch (1/2)
Passive Switch (2/2)

Active guidance!
Tilt- and Suspension Module

Dynamic decoupling of the shuttle body especially w.r.t. high frequencies

⇒ high comfort
On-Board Electrical System

Energieversorgungsmodul

- 550 - 750 V DC
- NiCd-Batterie
- DC/DC-Wandler
- 336 V
- 62,5 V
- Power Caps
- 24 V Versorgungsspannung:
  - Sensorik und Messverstärker
  - Steuerungshardware
  - aktive Kühlung

Hydraulikaggregat

- Fahrmodul 1
- Läuferwicklung 1
- Läuferwicklung 2
- Läuferwicklung 3
- Statorabschnitt

- Fahrmodul 2
- Läuferwicklung 4
Hydraulic System

Remark:

- Lost of pressure
  ⇒ no hydraulic actuator works!
  ⇒ automatic emergency braking
(C) Software Tasks

Within a shuttle:
- Control Traction Module
- Control Guidance module
- Control Tilt- and Suspension Module
- Control Energy Management
- Control Hydraulic System
- **Condition Monitoring**
- On-Board information system

Beyond one shuttle:
- Track Shuttle coordination for the linear drive
- Shuttle coordination (convoys, switches, …)
- Logistics
- …
Software Structure
Software Structure Guidance Module
Software Structure
Tilt- and Suspension Module
Shuttle Coordination

Fahrmodul

- Nickregelung
- Energiemanagement Shuttle
- Regelung der Position, Geschwindigkeit und Schubkraft
- Sensorik - Strom - Position - Luftspalt
- Spannungsversorgung der Läufer
- Energieversorgungsmodul

Leitwarte

Bediener:
Vorgabe von Sollposition und Referenzwerten für spezielle Manöver (Konvoifahrten)

Kommunikation über Funksysteme

Statorabschnitt 1
- Spannungsversorgung
- Statorstromregelung und digitale Informationsverarbeitung

Statorabschnitt 2
- Spannungsversorgung
- Statorstromregelung und digitale Informationsverarbeitung

Statorabschnitt 3
- Spannungsversorgung
- Statorstromregelung und digitale Informationsverarbeitung

Kommunikation Synchronisation über Feldbus

Statorregelung
Shuttle Coordination

- Planned sensors
Challenge: Safe Behavior

Background:
- Ensure safe operation is of paramount importance
- Operation must be safe also in the presence of hardware faults

State of the Art:
- Testing:
  - Executing the system to detect safety violations.
  - Scalable but incomplete (too low coverage)
    - not sufficient
- Modelchecking:
  - Complete formal verification of given safety properties
  - Not scalable and only applicable for abstract models
    (static structure/behavior; only finite Models)
    - not directly applicable
SFB614 Challenge: Self-Optimization

The system optimize its behavior on multiple levels:

- Logistics: maximize economical benefit, customer satisfaction, …
- Convoys, section control: minimize the energy costs for each convoy, minimize blocking, …
- Energy management: minimize energy costs for each shuttle, maximize battery life time, …
- Suspension/tilt modules, track control: drive comfort, driving safety, abrasion, …
- Actuators/sensors: abrasion, exact reaction, …

⇒ Reconfiguration at multiple levels at run-time
General Challenge: Multiple Domains

- **Domains:**
  - Logistic
  - Real-time coordination
  - Local control
  - Electronics
  - Mechanics

⇒ A more seamless integration between the different worlds is required!
I Introduction

I.1 Definitions & Terminology
I.2 Examples
I.3 System Characteristics
I.4 Application Domains
I.5 Resulting Challenges & Future Challenges
I.6 Discussion & Summary
I.7 Bibliography
I.3 System Characteristics

(1) Reactive systems
(2) Real-time systems
(3) Continuous/discrete/hybrid systems
(4) Embedded systems
(5) Dependable systems
(6) Distributed systems
(1) Reactive Systems

Transformational:
- output = f(input)
- Terminating

Reactive:
- Continuously interacting with the environment
- Non terminating
- Stimulus/response interaction (must be able to respond to interrupts)
(2) Real-Time Systems

- In a real-time system the timely response to environment stimulus is essential.

  ![](image)

  **Deadline:** the latest time until which the response to a stimulus is required.

  ![](image)

- **Synchronous/periodic task:** Can be used to address required activities with a period $p$. In the worst case, the response time $2p$ can be guaranteed by scheduling analysis (if preemption is possible, otherwise $p$).

  ![](image)

- **Asynchronous/aperiodic task:** Can be used to address required activities for stimulus without knowledge about the occurrence.
Hard and Soft Real-Time

- In a real-time system not only the correct functional behavior is relevant but also its timely provision.
- The utility describes the time-dependent value of a result.

**Soft real-time systems:** A failure to meet a specified deadline reduces the utility of the result, but does not lead to a significant financial loss (E.g., letter sorting machine)

**Hard real-time system:** A failure to meet a specified deadline can lead to catastrophic consequences (E.g., a computer system controlling a railway-crossing)
(3) Continuous/Discrete/Hybrid systems

- Continuous/discrete in time:

- Continuous/discrete in space/data:
Continuous/Discrete/Hybrid systems

- **Continuous:**
  - required input/output behavior is specified using differential equations
  - In practice numerical solver or difference equations
  - usually only quasi-continuous in time and data

- **Discrete:**
  - State transition systems (State machines, Petri-nets, …)
  - Complex high-level programs

- **Hybrid:**
  - Continuous and discrete elements
  - Either independent or interrelated (the hard case)
(4) **Embedded Systems**

- Software **embedded** into a **technical system** that interacts with the real physical world, controlling some specific hardware.

- Interaction with the technical system via **actors** and **sensors**

[Broekman&Notenboom2003]
Extra Hardware

- **Real-time clocks**: highly accurate elapsed time ticks (not calendar time!)
- **Interrupt controller**: hardware support for asynchronous stimulus (e.g., aperiodic tasks)
- **Hardware timers**: more flexible than RTOS ticks
- **Watchdog timer**: a one-shot hardware timer that results in a non-maskable interrupt when not retriggered.
- **A/D and D/A converter**: special support for converting analogous external signals into digital ones (and vice versa)
- **Serial communication controller**: often at least used for development and debugging
- **Bus controller**: direct support for special bus systems (e.g., controller area network (CAN))
Possible Hardware Platforms (1/2)

[Diagram showing possible hardware platforms with categories such as general-purpose computing and specialized computing, including microprocessors, microcomputers, etc.]

[Cooling2003, Fig. I.17]
Possible Hardware Platforms (2/2)

- **General purpose microprocessor**: X86, PowerPC, … + processor board
- **Highly integrated microprocessor**: additional I/O on the chip
- **Single-chip microcomputer**: I/O, Rom, RAM, …
- **Single-chip microcontroller**: microcomputer with real-time clock, A/D and D/A converters, …
- **Digital signal processor**: extremely high throughput, optimized for numerical operations
- **Mixed-signal processor**: direct interface to analogous and digital signals (usually low-cost)
- **Bespoken System-on-chip design**: FPGAs or ASICs that may incorporate microprocessors and memory (very expensive)
 Dependable Systems

Dependability is that property of a computer system such that reliance can justifiably be placed on the service it delivers.

[J.C. Laprie]

Dependability (Verlässlichkeit):

is defined as the trustworthiness of a computer system such that reliance can justifiably be placed on the services it delivers.
Dependable Systems

Two aspects of *reliance*:

(1) reliance that the system performs according to its service specification

(2) reliance that the system avoids hazards, i.e., behaviour which may lead to undesired consequences
Attributes of Dependability

- **reliability** (Funktionsfähigkeit)
  dependability with respect to continuity of service
- **availability** (Verfügbarkeit)
  dependability with respect to readiness for usage
- **safety** (Sicherheit)
  dependability with respect to avoidance of catastrophic consequences
- **security** (Vertraulichkeit)
  dependability with respect to prevention of unauthorized access and/or handling of information
Reliability

Reliability is the probability of a component, or system, functioning correctly over a given period of time under a given set of operating conditions.

Quantitative:

Reliability $R(t)$ is the probability that the system will conform to its specification throughout a period of duration $t$.

- Note that $t$ is important
- If a system only needs to operate for ten hours at a time, then that is the reliability target.
Availability

The availability of a system is the probability that the system will be functioning correctly at any given time.

Quantitative:

**Availability** $A$ (or $V$) is the percentage of time for which the system will conform to its specification.

- Literally, readiness for service
Safety

- **Safety** is a property of a system that it will not endanger human life or the environment.

- A safety-related system (safety-critical system) in one by which the safety of equipment or plant is assured.

[Storey1996]
Security

- **Security**: Prevention of or protection against (a) access to information by unauthorized recipients or (b) intentional but unauthorized destruction or alteration of that information.

Security vs. Safety:

- Risk for privacy, (organisational privacy,) or national security
- When unauthorized access to a system can result in an accident \( \Rightarrow \) (safety requires some security)
- When any unauthorized disclosure is considered to be an accident \( \Rightarrow \) (security \( \subseteq \) safety)
(6) Distributed Systems

A collection of autonomous computers linked by a computer network, and communicate and coordinate their actions only by message passing.

- It has the following characteristics:
  - concurrency
  - lack of global clock
  - independent failure

[Coulouris+2004]
Challenges of Distributed Systems

- Heterogeneity
- Resource Sharing
- Openness
- Concurrency
- Scalability
- Failure handling (fault tolerance)
- Transparency
- Security
I Introduction

I.1 Definitions & Terminology
I.2 Examples
I.3 System Characteristics

I.4 Application Domains

I.5 Resulting Challenges & Future Challenges
I.6 Discussion & Summary
I.7 Bibliography
I.4 Application Domains

- Automotive
- Transportation
- Avionics
- Space missions
- Medicine technique
- Industrial automation
- Telecommunication
Automotive

Characteristics:
- Hard real-time
- Sever resource constraints (high cost pressure)
- Highly interconnected
- High reliability and safety requirements
- User interface important!
- Extreme increase in complexity (can be observed and is further expected)

Trends:
- Drive-by-wire
- Model-driven development
- Test automation
- Software architecture (AUTOSAR)
- Safety analysis

[Liggesmeyer&Rombach2005, chapter 16]
[Schuffele&Zurawka2004]
http://www.heise.de/ct/03/14/170/
Transportation

Characteristics:
- Hard real-time
- Some resource constraints
- Large-scale systems
- Heterogeneous systems
- Very high safety requirements

Observation:
- Safety standards important (CENELEC, EN 50128)
- Very long-winded certification procedures (even for small components)
- Very conservative approach common

[Liggesmeyer&Rombach2005, chapter 17]
Avionics

Civil aircrafts
Characteristics:
- Hard real-time
- Some resource constraints
- Very high safety requirements

Observation:
- Safety standards important (DO178B, …)
- Certification only for the whole plane!
- Some early adaptors of formal methods (Esterel; SCADE; …)

Military avionics systems
Should be:
- Unmanned/Autonomous
- Adaptive,
- fault-tolerant

Remark: different goals!
Space Missions

Manned missions
Characteristics:
- Hard real-time
- Some resource constraints (weight)
- Extreme network delays!
- Extreme safety requirements
- Mission time counts!

Observation:
- Special safety standards
- Severe cost pressure

Unmanned missions
Should be:
- Unmanned/Autonomous
- Adaptive,
- fault-tolerant

[Liggesmeyer&Rombach2005, chapter 18]
Medicine Technique

Characteristics:
- Safety is preliminary concern
- Ergonomics
- Robustness
- Security
- Hard real-time
- Some resource constraints

Observation:
- Specific standards
- Liability issues make online recording of data an attractive option (replay scenarios)

[Liggesmeyer&Rombach2005, chapter 19]
Industrial Automation

Characteristics:

- Large-scale hierarchical systems:
  - Administration (finances, human resources, documentation, long-term planning)
  - Enterprise (set production goals, plans enterprise and resources, coordinate different sites, manage orders)
  - Production (Manufacturing, Supervision)
  - Automation (Group (Area), Unit (Cell), Field)

- Heterogeneous systems

- High safety requirements

- Very high reliability and availability requirements

- Only minor resource constraints (PLCs)

- Hard real-time (often addressed by robust designs)

Observation:

- Very expensive components

- Fail Safe were possible

[Liggesmeyer&Rombach2005, chapter 20]
Telecommunication

Characteristics:
- Large-scale systems
- Very high availability requirements (usually about 99,999%)
- Only minor resource constraints
- Heterogeneous systems
- Many features (feature interaction)

Observation:
- Early adopters for model-based development and automatic code generation (SDL)
- Software architecture
  - Performance
  - Protocol selection
  - Failure management
  - Separation of features

[Liggesmeyer & Rombach 2005, chapter 21]
I Introduction

I.1 Definitions & Terminology
I.2 Examples
I.3 System Characteristics
I.4 Application Domains
I.5 Resulting Challenges & Future Challenges
I.6 Discussion & Summary
I.7 Bibliography
Software is the Bottleneck

In the (recent) past, an embedded system would be either small or simple, or the composition of almost non-interacting imported and assembled components. The trend is that the number and complexity of functions will increase drastically. Increasing complexity is making the present design methodologies rapidly obsolete. Productivity of the order of six (or less!) lines of embedded code per day per person is common in HRT embedded systems. The cost of developing a new plane (of the order of several billions of Euros) is about ½ related to embedded software and electronics subsystems.

[Bouyssounouse&Sifakis2005, p. 4]
Complexity of Design Flows and Supply Chains

The electronics industry is increasingly disaggregating: new opportunities are now opening up for subsystem and component suppliers. These dynamics are stressing the interfaces among the supply chain players. Several quality problems and time-to-market delays can be traced to specification and system integration difficulties. Among the most challenging supply chains to support are the automotive and avionics chain.

[Bouyssounouse&Sifakis2005, p. 8]
Need for Self-Adaptive/Self-Optimizing Behavior

In the near future, software-intensive systems will exhibit **adaptive** and **anticipatory behavior**; they will process knowledge and not only data, and **change their structure dynamically**. Software-intensive systems will act as global computers **in highly dynamic environments** and will be based on and integrated with service-oriented and pervasive computing.

[EU-NSF-SIS2004]
A New Software Design Paradigm

The highest ranked recommendation was to develop a new software design paradigm that recognizes that uncertainty and emergent, unanticipated behaviors are likely to be forever present in the software-intensive systems of the future due to their ultra-large, networked, distributed, and diffuse-control natures.

[NSF-DLS2003, Executive Summary]
Model-Based Development

A promising approach for solving these problems is **model-based development**: models can serve as a vehicle for communicating information between *business process engineers, control engineers* and *computer scientists*, and can also provide an additional basis for preimplementation validation of requirements and quality properties as well as for automatic generation of source code. However, this paradigm for software-intensive system development is still in its infancy [...].

[EU-NSF-SIS2004, p. 8-9]
I Introduction

I.1 Definitions & Terminology
I.2 Examples
I.3 System Characteristics
I.4 Application Domains
I.5 Resulting Challenges & Future Challenges
I.6 Discussion & Summary
I.7 Bibliography
I.6 Discussion & Summary

- A software-intensive system is any system where software contributes **essential influences** to the design, construction, deployment, and evolution of the system as a whole. Software-intensive systems include large-scale heterogeneous systems, embedded systems for automotive applications, telecommunications, wireless ad hoc systems, business applications with an emphasis on web services etc.

- Software-intensive systems are characterized by their reactive nature, real-time requirements, mix of continuous and discrete behavior (hybrid), the embedded character of some components of the system, the required dependability and the distribution of its elements.
Discussion & Summary

- The typical application domains for software-intensive systems are, for example, automotive, transportation, avionics, space missions, medicine technique, industrial automation, or telecommunication. These domains have quite different requirements and characteristics.

- The current challenge is to stay in control of the quality and time-to-market delays by means of specification and system integration techniques such as interfaces and components. Software-intensive systems of the future will due to their ultra-large, networked, distributed, and diffuse-control natures require that the systems are self-adaptive/self-optimizing and therefore configure themselves online. An envisioned solution is the model-based development of software-intensive systems which promise to address the integration as well as time-to-market problems encountered today.
I.7 Bibliography (1/4)


I.7 Bibliography (2/4)


I.7 Bibliography (3/4)


I.7 Bibliography (5/4)


