

Software Engineering for Software-Intensive Systems: VIII Summary and Outlook

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



VIII Summary and Outlook

VIII.1 Summary

VIII.2 Outlook

VIII.3 Bibliography



Who is involved in SIS?

Application Domains :

Automotive, Transportation, Avionics, Space missions,
Medicine technique, Industrial automation,
Telecommunication, ...

System Characteristics:

Reactive systems, Real-time systems,
Continuous/discrete/hybrid systems, Embedded systems,
Dependable systems, Distributed systems

Involved disciplines (need for integration):

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



How can we model SIS?

Structure:

Block Diagrams, Class Diagrams, Component Diagrams, ...

Behavior:

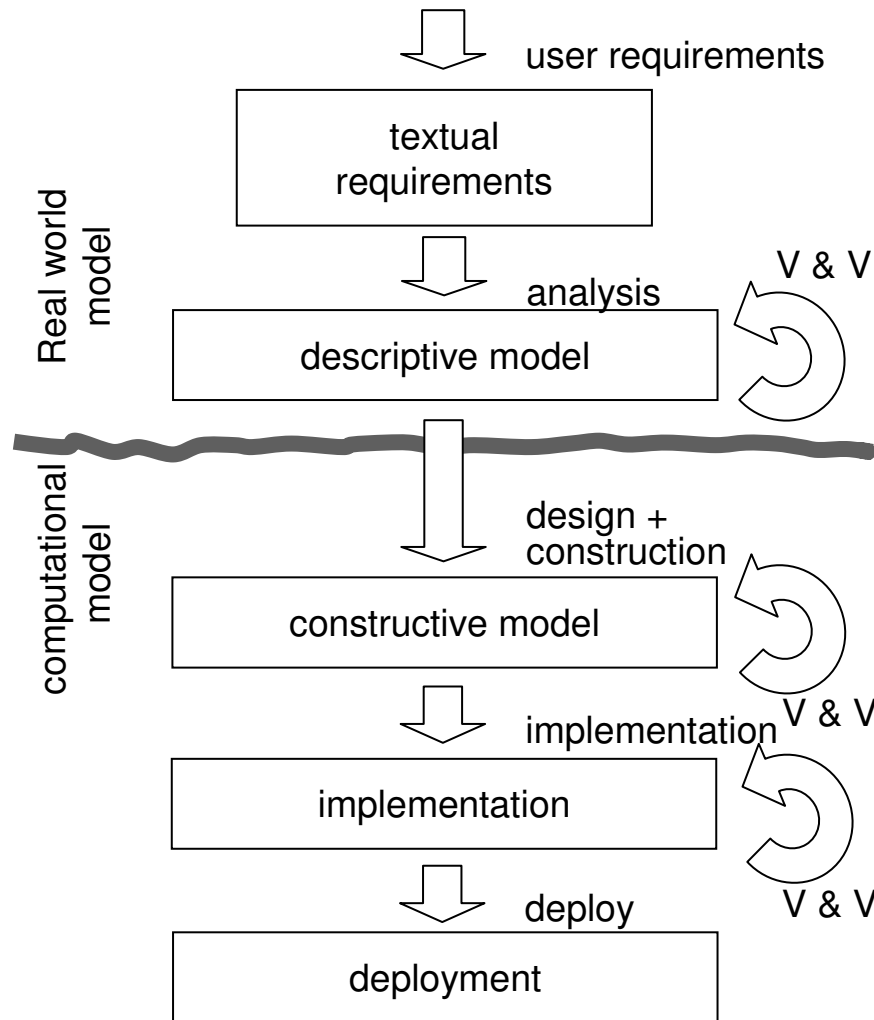
Automata/State Machine, Timed Automata, Continuous Behavior, Hybrid Automata, Hybrid Reconfiguration Automata, Petri Nets, Graph Transformation Systems

Properties:

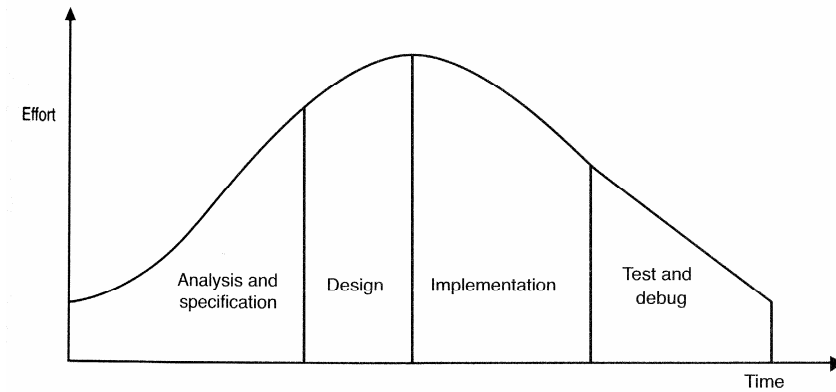
Safety (nothing bad happens), liveness (something good happens), stability (small initial differences \Rightarrow small differences over time), controllability (ability to move a system to a required state), observability (can the internal state be inferred by knowledge of its external outputs)



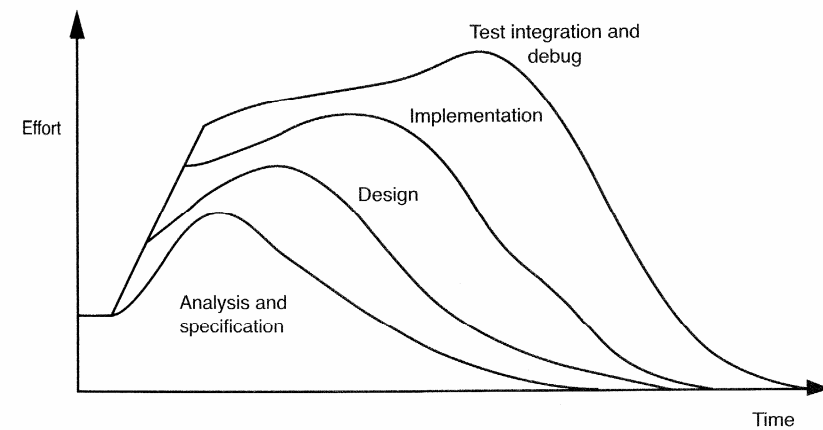
How are the different activities related?



[Cooling2002]



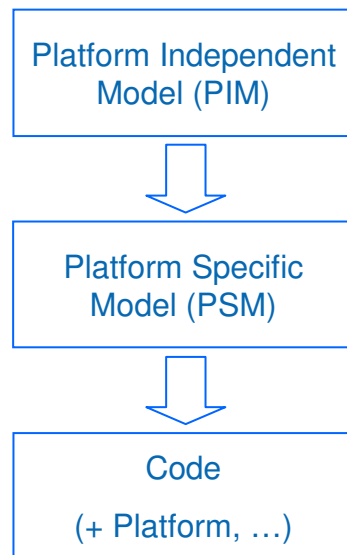
[Cooling2002]



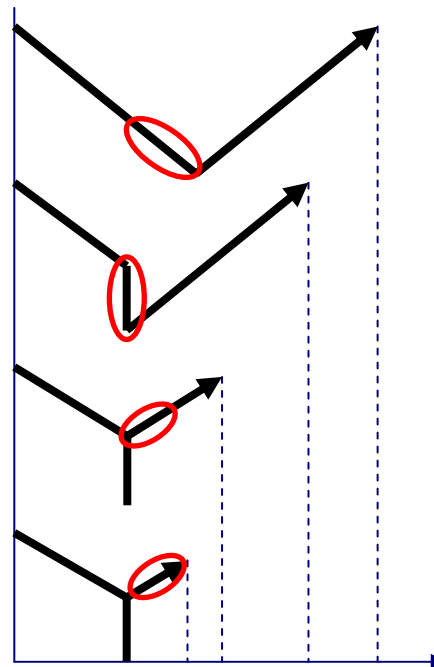


How does the process look like?

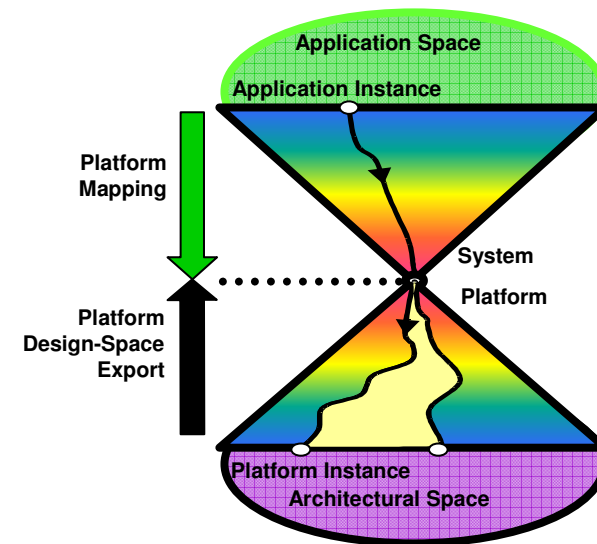
Current Proposals:



MDA



Y-Model



Platform-Based Design



What can be used for Requirement?

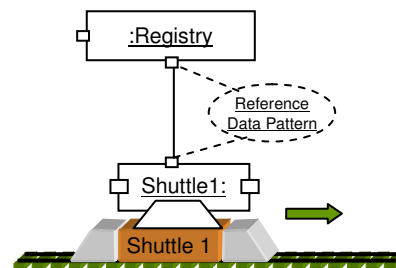
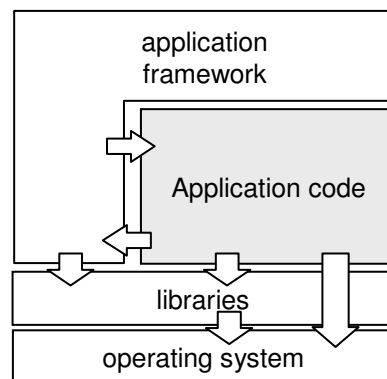
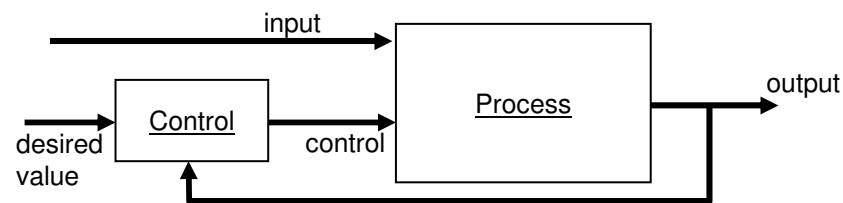
- Textual Document
- Requirement Diagrams (SysML)
- Use Case Diagrams (UML/SysML)
- Scenarios: SD or AD (UML/SysML)
- Hierarchies of goals
- ...



What can be used for Analysis/Design?

Analysis:

- Functional structuring
- Data flow structuring
- Object structuring



Design:

- Functional structuring
- Data flow structuring
- Object structuring
 - Components (explicit interfaces & independent deployable)
 - Agents (autonomous, situated in context, reactive, proactive)

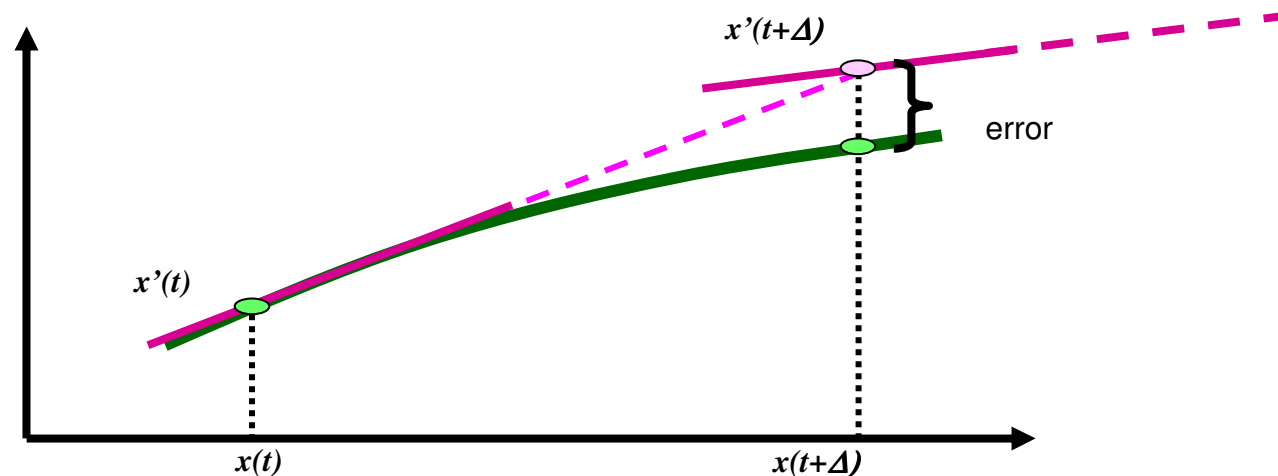
Advanced Design Concepts

- Software Pattern
- Software Architectures
- Toolkits & Frameworks
- Mechatronic UML



How can we implement it?

- **Discrete behavior (untimed):** trivial
- **Discrete behavior (timed):** either synchronous hypothesis or synthesis problem (period)
- **Continuous behavior:** approximation via Euler, Heun, Runge-Kutta, ...





How can we guarantee Quality (V & V)?

Life cycle phase	Dynamic analysis	Static analysis	Modelling
Requirements		X	X
Top-level design		X	X
Detailed design		X	X
Implementation	X	X	
Integration testing	X	X	X
System validation	X		X

[Storey1996]



VIII Summary and Outlook

VIII.1 Summary

VIII.2 Outlook

VIII.3 Bibliography



VIII.2 Outlook

- Software is today the **bottleneck** in productivity (especially for embedded systems)
 - ⇒ **Model-driven development** and improved **automation**
- **Complex supply chains** have to be supported
 - ⇒ Standardization and more powerful notions of **interfaces** (timing, safety, resource usage, ...)



Outlook

- Act in **highly dynamic environments**
⇒ **adaptive** and **anticipatory** behavior; change their structure dynamically.
- **Networking** with other systems and service-oriented and pervasive computing
⇒ Concepts for the integration of **local** and **distributed** processing are required!
- Hybrid Behavior is often **not robust**
⇒ Improved **analysis techniques** are required!

Finally: Many competent SWE people are required!



VIII.3 Bibliography

[Cooling2002] Jim Cooling, Software Engineering for Real-time Systems. Addison Wesley, November 2002

[Storey1996] N. Storey. Safety-Critical Computer Systems. Addison-Wesley, 1996.