Embedded Software Laboratory

RWTH AACHEN UNIVERSITY

Hardware Platform Design Decisions in Embedded Systems - A Systematic Teaching Approach -

Falk Salewski, Stefan Kowalewski

- WESE 2006 -

Motivation

- Embedded Systems = many different types of hardware platforms
  - CPU-based systems:
    - Microcontroller (MCU)
    - Digital Signal Processor (DSP)
    - Programmable Logic Controller (PLC)
    - …
  - Programmable Logic Devices (PLDs):
    - Field Programmable Gate Arrays (FPGA)
    - Complex Programmable Logic Arrays (CPLD)
    - …
- Platforms have several differences (internal structure, design process)
- Which platform is suited best? ➔ selection by expert in industry

➔ How we can provide students with a framework for a systematic hardware selection?

Teaching embedded systems (CS)

Winter	Introduction to Embedded Systems	Dynamic Systems for Computer Scientists	Lab Course
Summer	Embedded Software Design	Safety & Reliability Engineering	Formal Methods for Embedded Systems

- Lectures + exercises give an introduction in
  - Hardware platforms
  - Design processes
- Lab course imparts practical knowledge and programming skills
  - Different hardware platforms clarify (functional) differences
- Remaining question: When to use which hardware platform?

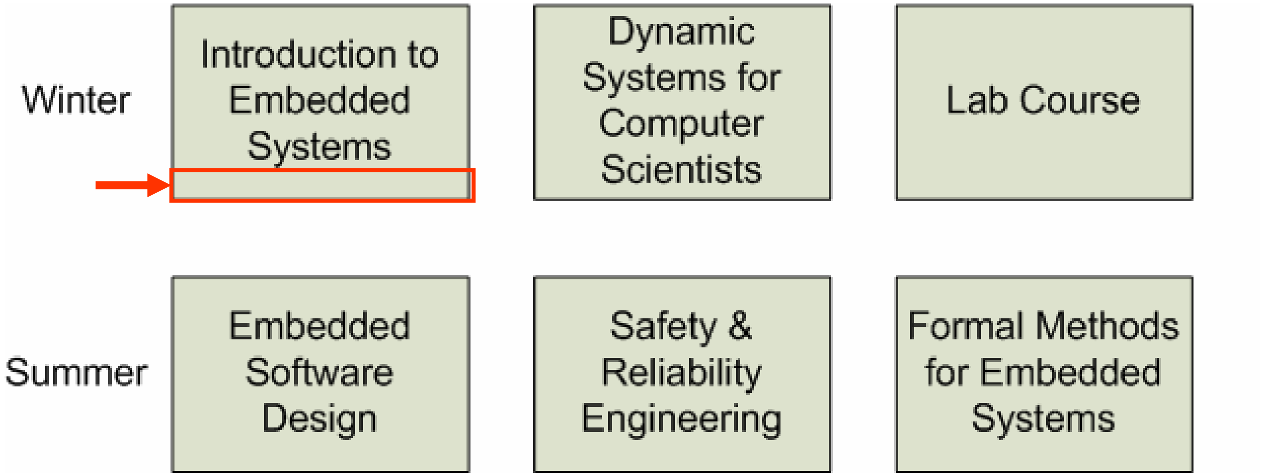
Embedded Software Laboratory

RWTH AACHEN UNIVERSITY

Systematic hardware selection

- What we already teach
  - Properties of different hardware platforms
  - Requirements engineering for embedded systems
  - Design processes in embedded systems
- What we need in addition
  - Capability to systematically analyze both functional and non-functional features of hardware devices
  - Skills to map these features to the requirements of a given specified application
  - An approach including hardware **and** software issues
  - Consideration of the fact that available hardware devices are constantly changing over time

Teaching embedded systems (CS)

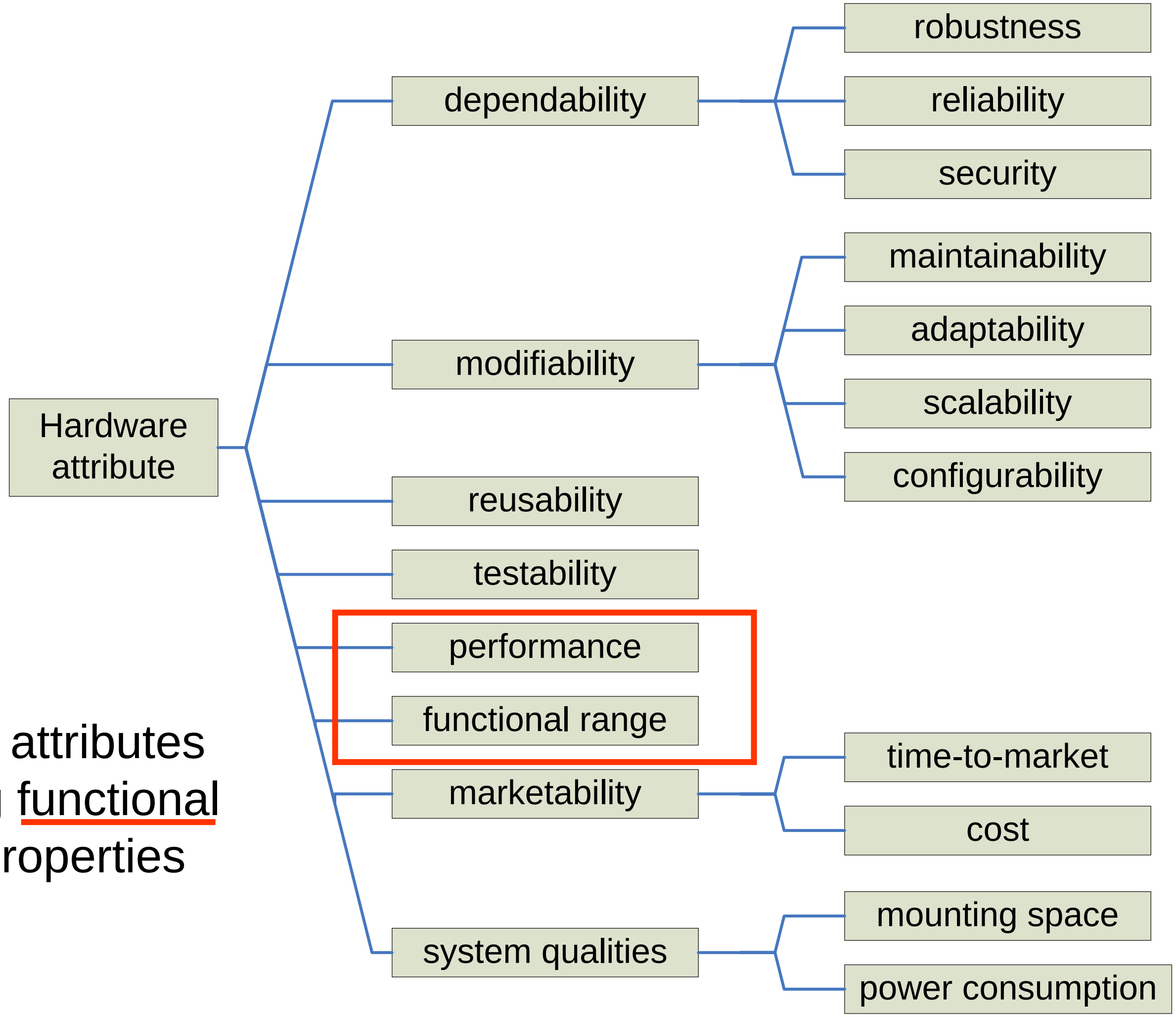


- Framework for systematic hardware selection at the end of the introductory course
- Next:
  - Presentation of our approach
  - Integration in the education

System requirements ⇔ hardware properties

- How to map hardware properties to system requirements?
- System Quality 🡐 HW attributes 🡐 HW properties (features)
- First step
  - Summarize all hardware attributes (functional and non-functional) which influence system's qualities
  - Example hardware attribute: *robustness* of hardware platform
- Second step
  - Summarize all hardware properties which influence hardware attributes
  - Example hardware property: *protective I/O circuits* influence the attribute *robustness* of a hardware platform

First step: Hardware attribute tree

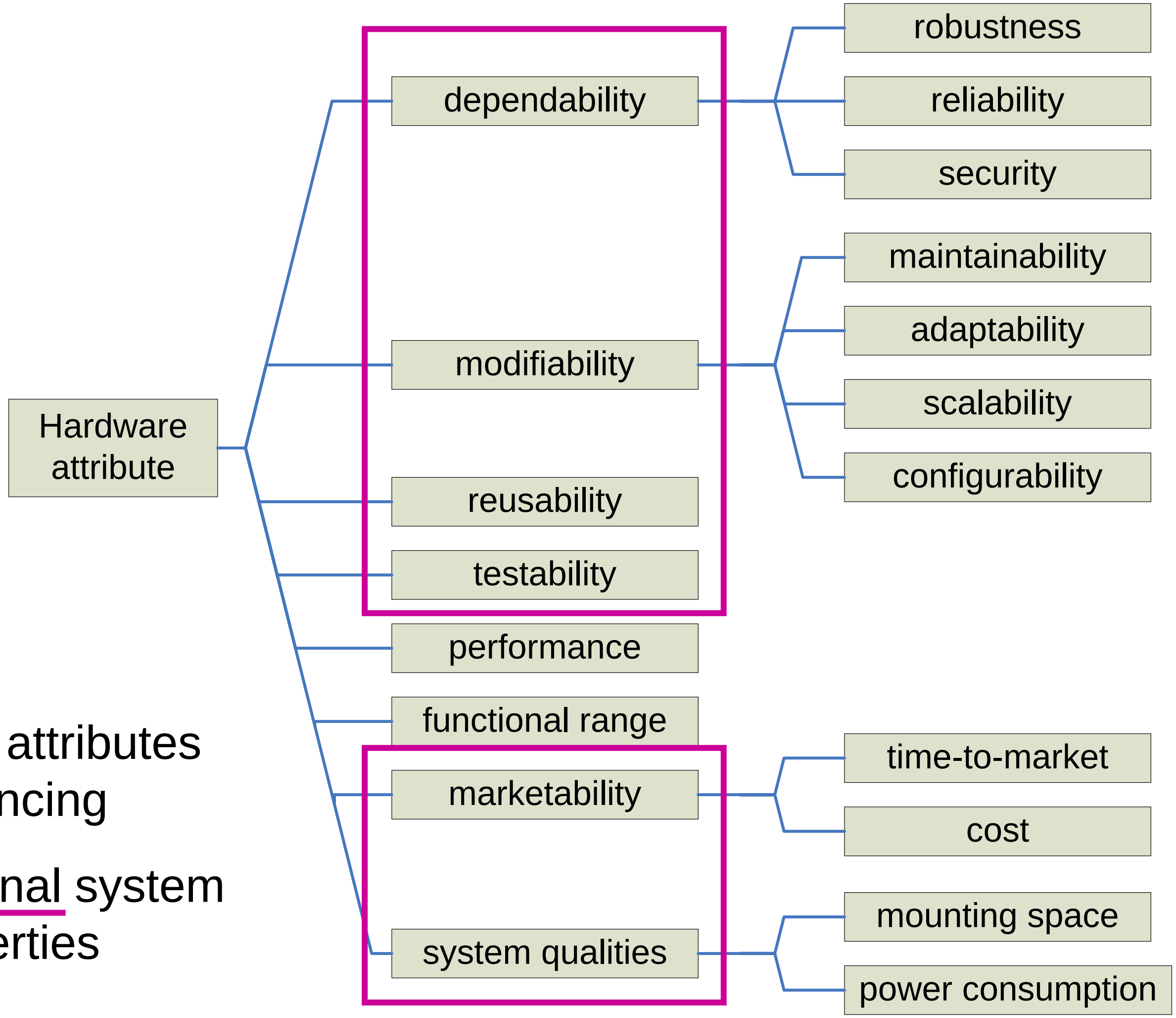


Hardware attributes influencing functional system properties

Embedded Software Laboratory

RWTH AACHEN UNIVERSITY

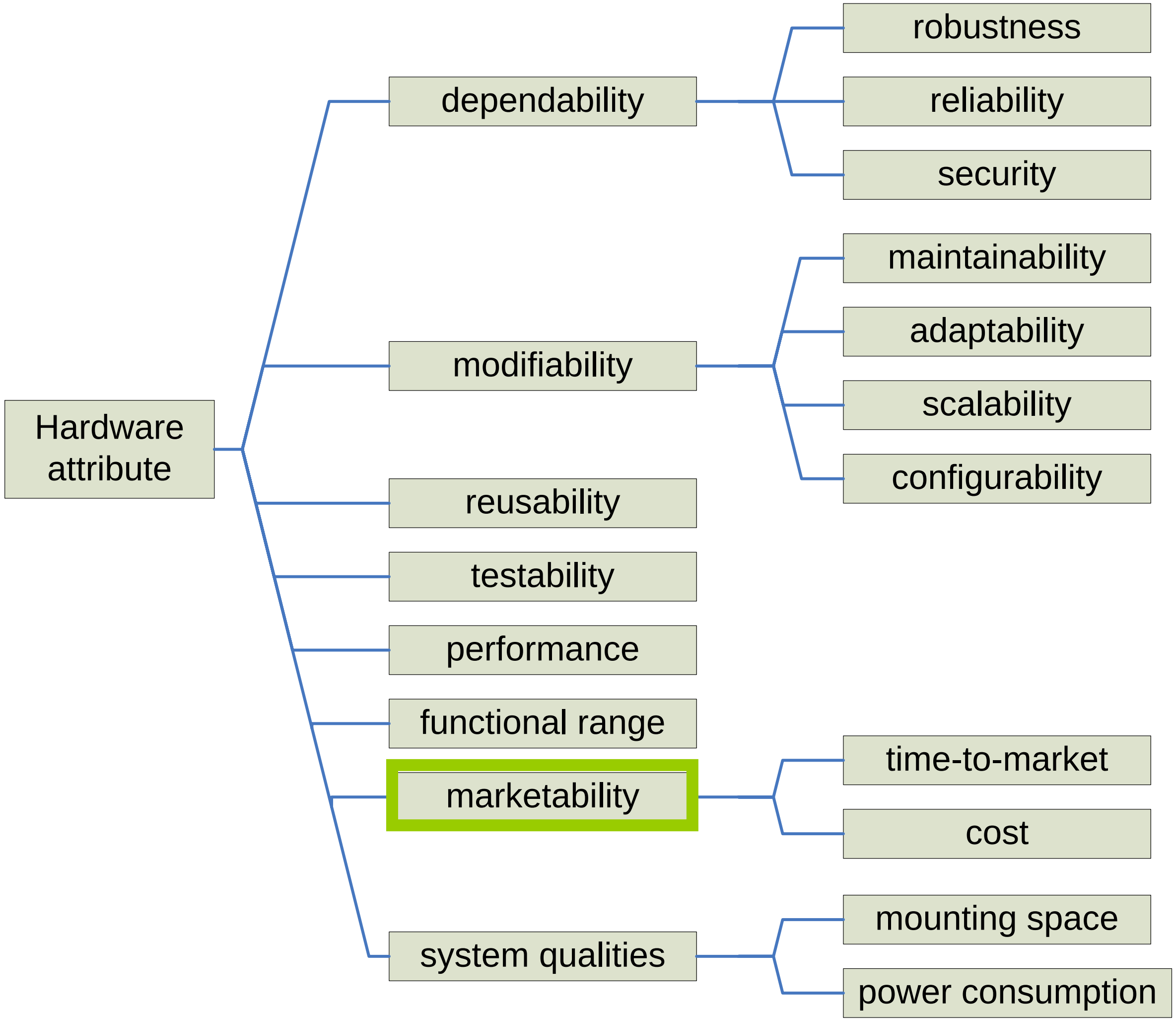
First step: Hardware attribute tree



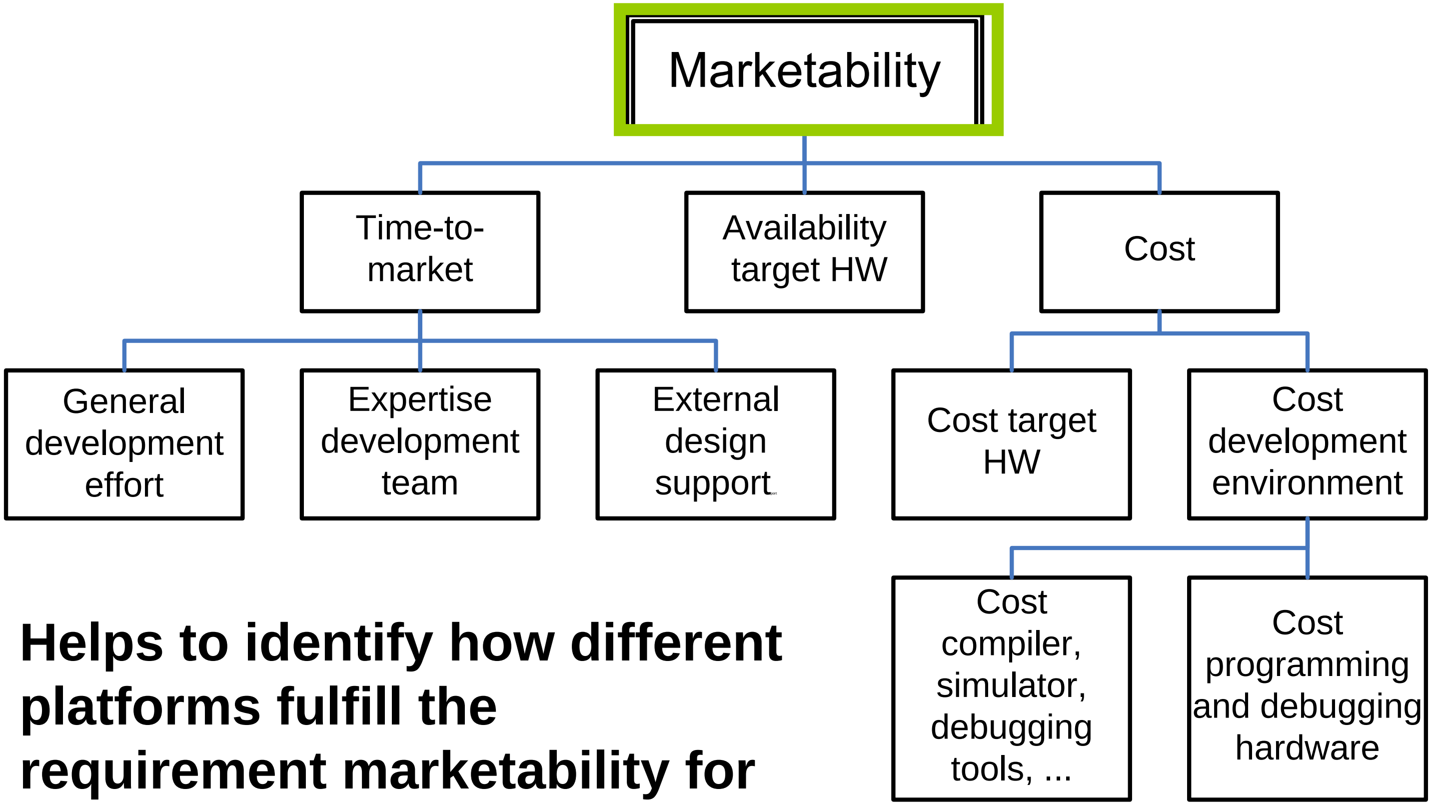
Hardware attributes influencing

non-functional system properties

2nd step: Influencing HW properties

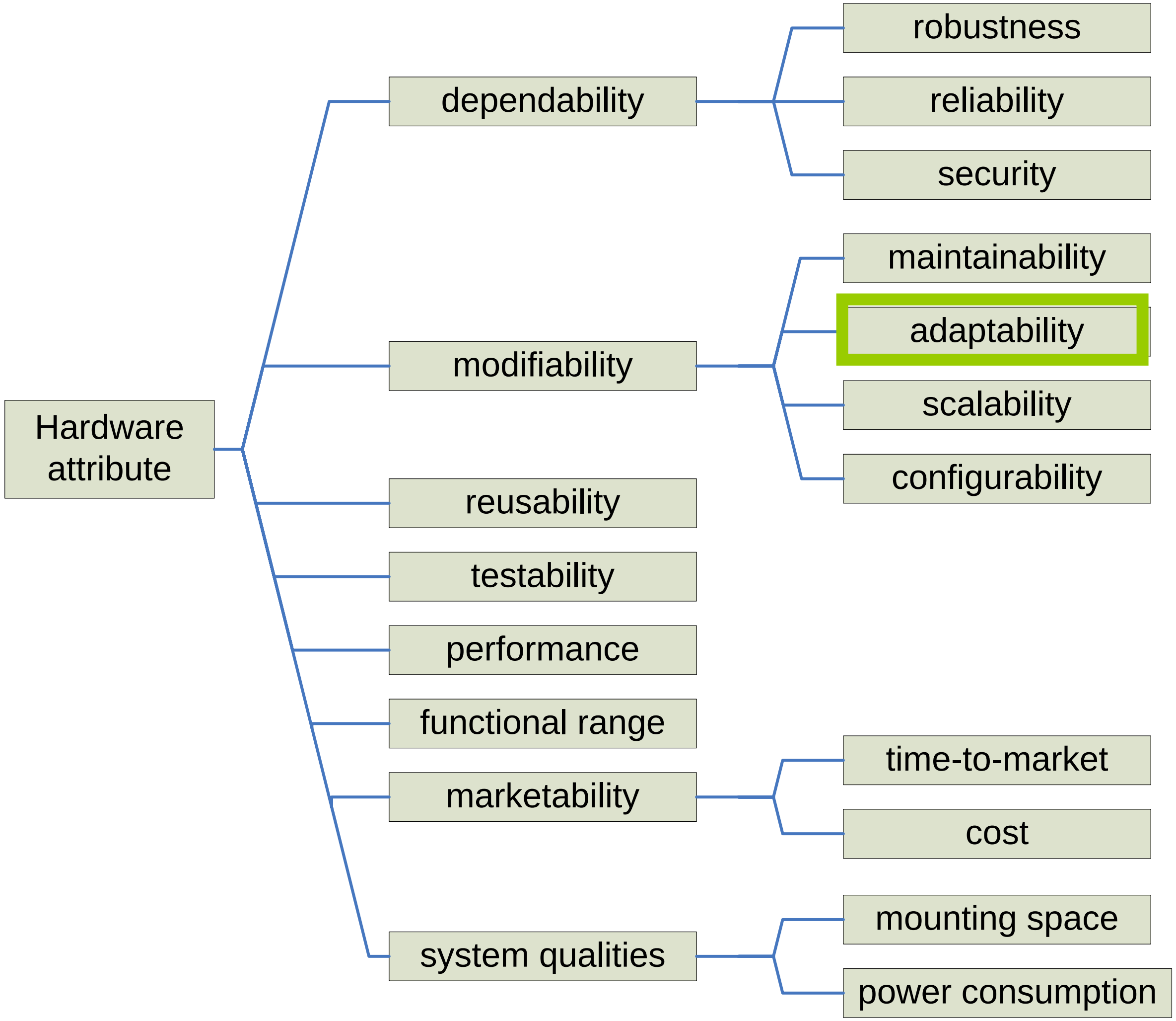


HW properties → marketability (abstract)

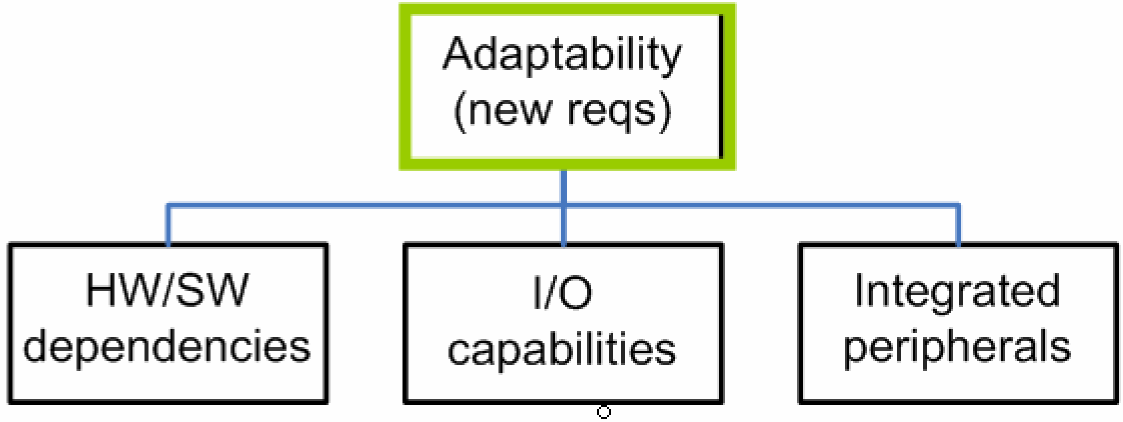


Helps to identify how different platforms fulfill the requirement marketability for the desired application

2nd step: Influencing HW properties



HW properties ➔ adaptability



next

Details: HW/SW dependencies ➔ adaptability

Factor influencing adaptability	CPU	PLD	Description, Details
HW/SW dependencies	X	X	Determines the effort necessary to transfer SW from one device to another (transfer is necessary if requirements cannot be fulfilled with the actual device)
	X		*MCU families* ease migration from one MCU to another of the same family (often special migration notes available)
	X		*Hardware abstraction/OS* could decrease dependencies between HW and higher SW layers
		X	If functionality is described in HDL (behavioral), a module can be transferred easily to any PLD suitable (new pin assignments necessary)
		X	If functionality is described in structural description, the module can be transferred easily only to hardware platforms with a similar structure (e.g. same basic elements)

back

Details: I/O capabilities ➔ adaptability

Factor influencing adaptability	CPU	PLD	Description, Details
I/O capabilities	X	X	Determines how easy new requirements with respect to I/O pins could be realized
	X		Usually, certain I/O functionality is mapped to a particular I/O pin Some I/O functionalities can be mapped to different I/O pins Few MCUs offer a free mapping of functionalities to I/O pins
	X		External busses (system bus, SPI, I2C, …) ease the integration of additional external peripherals/memory in the system
		X	Usually, all I/O pins have the same properties/options (e.g. termination, voltage level,…) Clock signals should be fed into the device via dedicated I/O pins
		X	Mapping of functionalities to I/O pins is done by SW Pin assignment could influence amount of chip area used for the design

back

Details: Integrated peripherals ➔ adaptability

Factor influencing adaptability	CPU	PLD	Description, Details
Integrated peripherals	X	X	Multi purpose integrated peripherals increase adaptability
	X		Functionality of integrated peripherals can be determined via dedicated registers
	X		Integrated peripherals with a high number of options increase adaptability (and complexity)
		X	Almost all (digital) functionality is determined via software Predesigned modules are available for common functions, written in HDLs (soft cores)
		X	Hard wired peripherals (e.g. clock divider) can be used if available Integrated peripherals usually do not include analog-to-digital

back

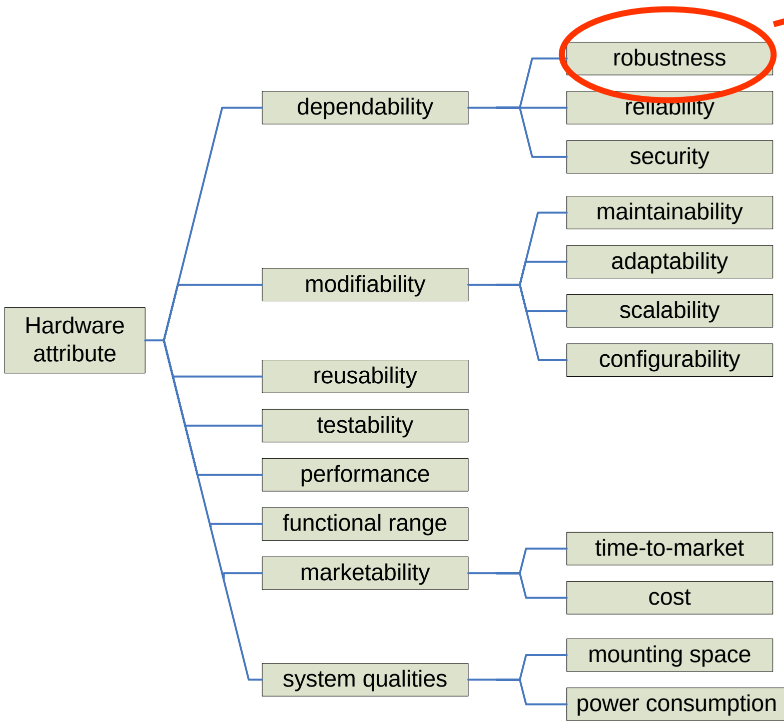
Selection process

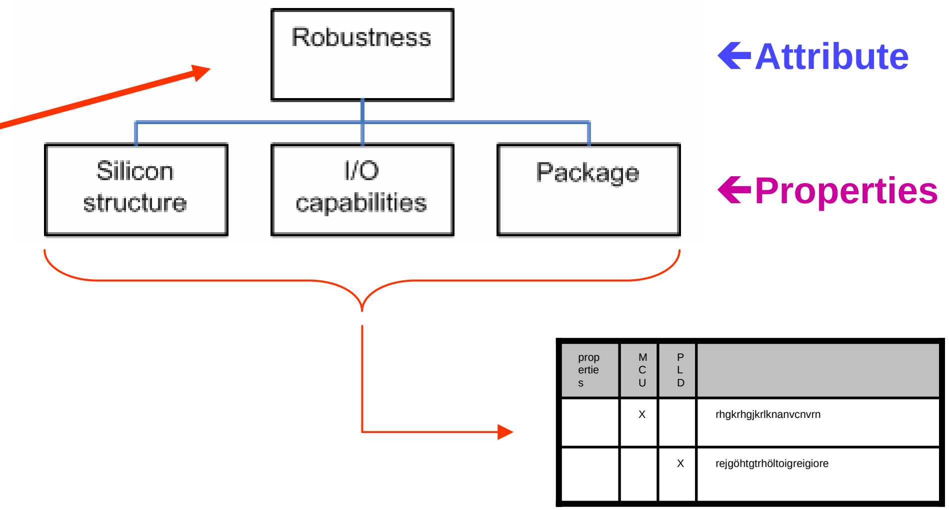
- Requirements engineering process
  - Determine functional and non-functional requirements of the overall system
  - e.g.: robustness: ++ (very important)
    adaptability: +/- (nice to have)

- Analysis of different hardware platforms
  - How do certain HW platforms fulfill system requirements?

Selection process (Analysis)

- Analysis:





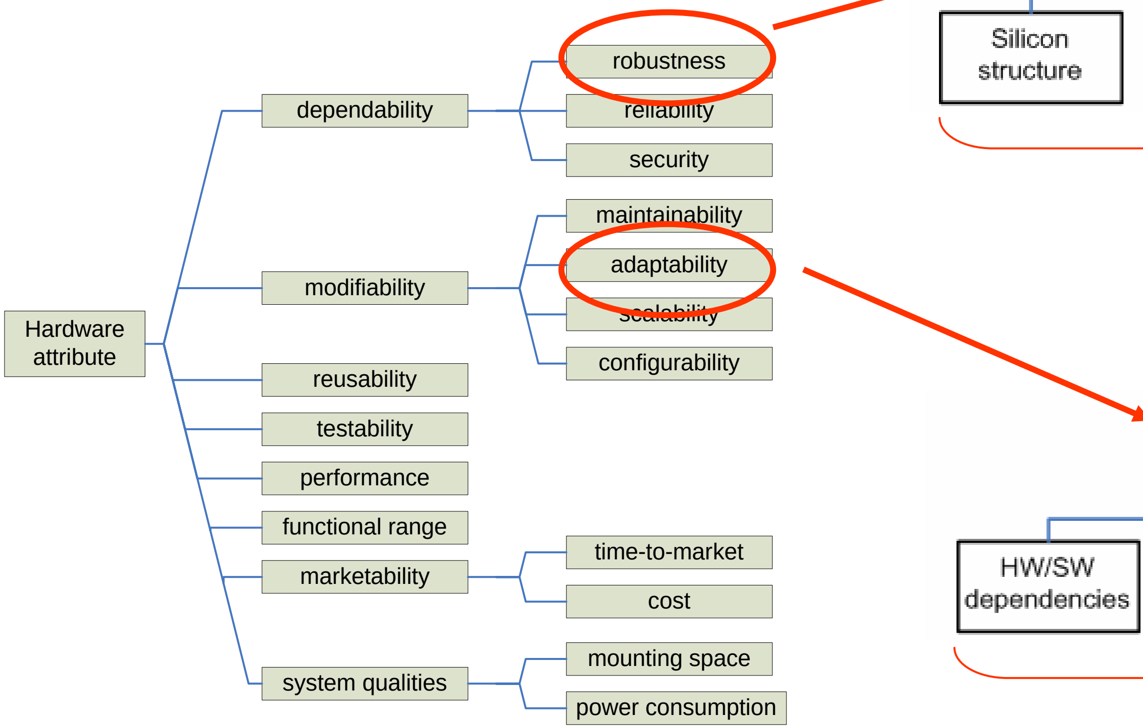
- Analyse hardware properties of different platforms
- Use this information to rate the corresponding hardware attribute for each platform

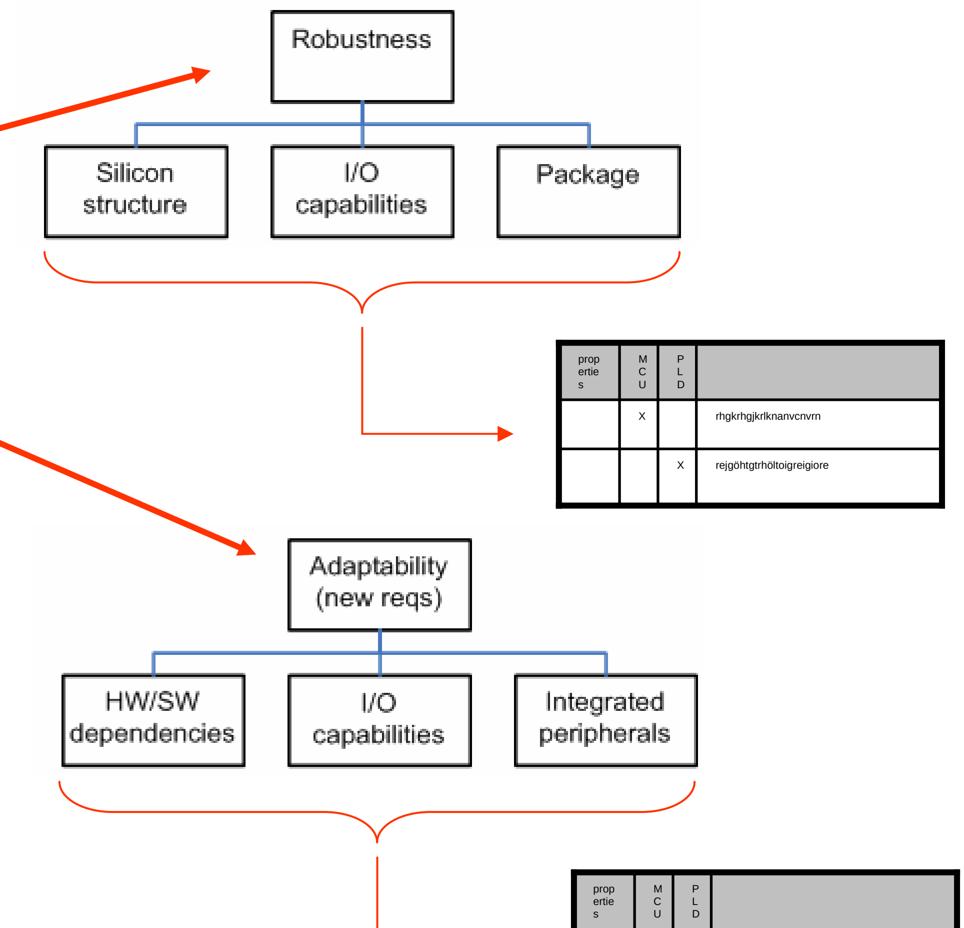
Embedded Software Laboratory

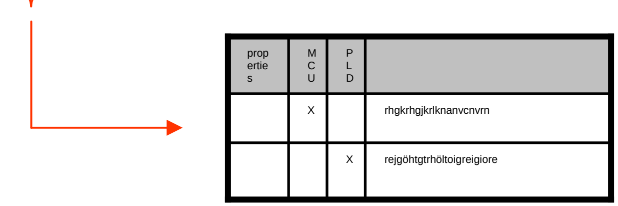
RWTH AACHEN UNIVERSITY

Selection process (Analysis)

- Analysis:







➔Analyse & rate different hardware platforms

Selection process (cont.)

- Requirements engineering process
  - Determine functional and non-functional requirements of the overall system
  - e.g.: robustness: ++ (very important)
    adaptability: +/- (nice to have)

- Analysis (two step approach)
  - How do certain HW platforms fulfill system requirements?

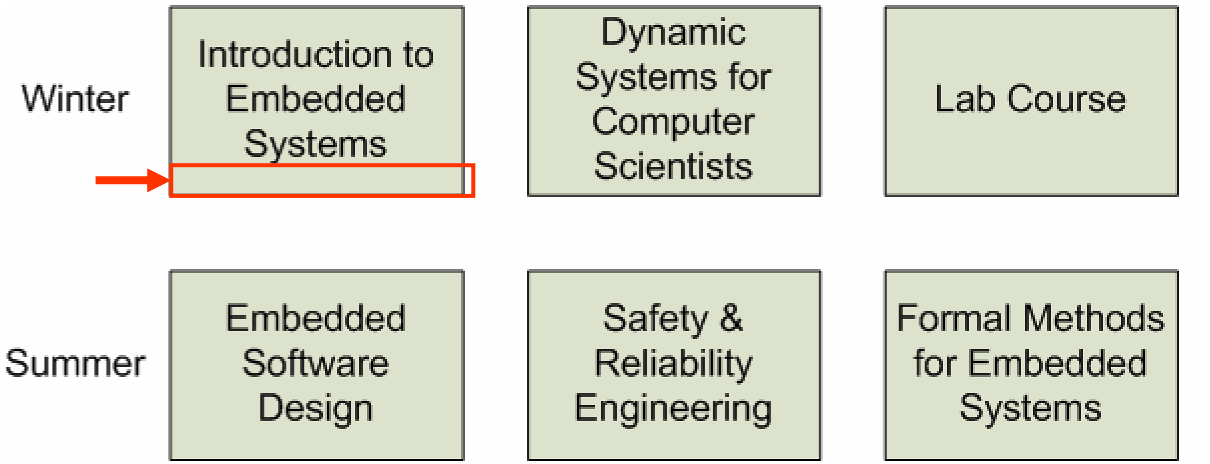
    ➔ Rate hardware attributes for each hardware platform
  - e.g.:

HW1:	robustness:	++	adaptability:	+
HW2:	robustness:	+/-	adaptability:	++

- Selection on basis of rating

  ➔ Usually, trade offs are necessary (example in the paper)

Integration into ES education



- Integration at the end of the introductory course
  - Present hardware attribute tree
  - Develop structure „influencing hardware properties“ with students
- Advantages:
  - Development with students allows repetition/survey of lecture contents
  - Students learn a systematic selection process

Conclusions

- Need for educating systematic hardware platform selection
- 2-step approach for analysis
  - Hardware attribute tree
  - Hardware properties influencing these attributes
- This approach allows
  - Students with little experience in hardware platforms to understand and realize systematic hardware selection

    (use completed approach / use it as framework)
  - Repetition/survey of lecture contents if the development of the structures is done together with the students

    (develop approach)
- The approach is not device dependent and thus flexible for future devices

Future Work

- Evaluate and improve our approach in education
- Realize a web system which allows
  - An intuitive representation of the contents presented
  - Feedback from users
    - Students
    - Other institutes/universities
    - Industry
  - Integration of knowledge from companies developing/selling embedded hardware platforms

Hardware Platform Design Decisions in Embedded Systems - A Systematic Teaching Approach -

Falk Salewski, Stefan Kowalewski

- WESE 2006 -