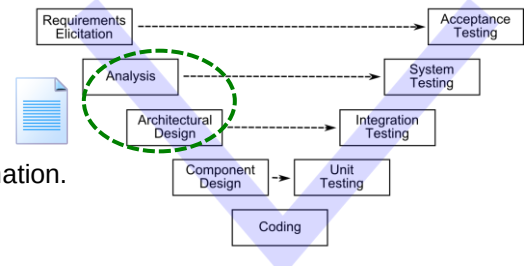
Exactly the Information your Subcontractor Needs: DeSyRe — Decomposing System Requirements

Dr. Birgit Penzenstadler
Universitat Polytècnica de Catalunya
Barcelona, 30. November 2011

1

Motivation: Distributed Development

- **Situation**:
  Company assigns the development of subsystems to subcontractors.
- **Problem**:
  Requirements specifications for subsystems often lack information.
- **Consequence**:
  Costly inquiries or later incompatibilities.

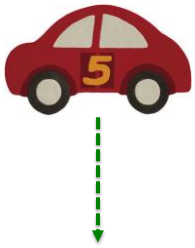


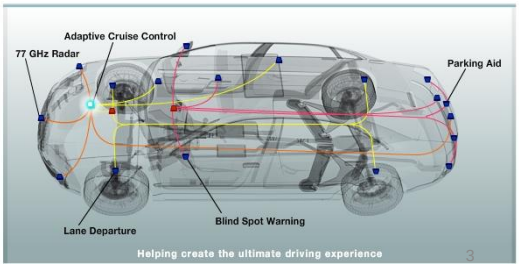
- **Reason**: Requirements deduction unclear.
- **State of the art**:
  Pragmatic handling in practice, island solutions in research.
- **Result**:
  Missing method for the systematic deduction of subsystem requirements.

2

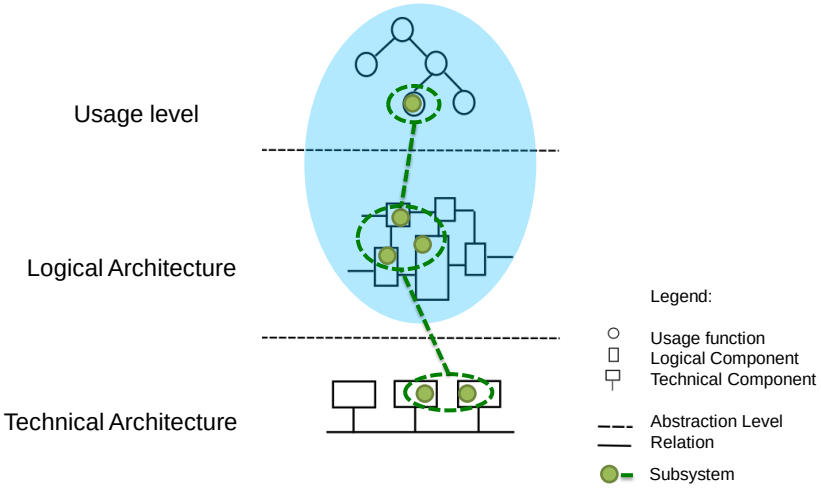
Motivation: Decomposition of Requirements

- **Example:**
  Development of a car,
  assignments to subcontractors.
- **Requirement**: (case study)
  „The velocity of the vehicle shall be automatically adaptable to the velocity of the preceeding vehicle.“
- **Question**:
  How do we have to decompose this requirement?



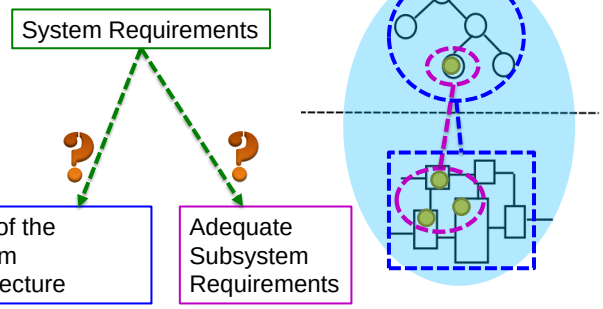


Foundation: Examination using abstraction levels



4

Challenges & Contributions



Challenges:

Draft of the System Architecture

Adequate Subsystem Requirements

Contributions:

- Catalogue for Decomposition Criteria
- Interview Study on System Decomposition

- Patterns for Requirements Decomposition
- Method for Implementation
- Case Studies on Applicability and Usefulness

5

Interview Study

- Participants: 7 software and system developers from the OEMs BMW, Daimler, Audi and MAN, from the subcontractors Bosch and Siemens VDO (later on Continental) and Berghof Automation
- Purpose (template acc. to Runeson, Wohlin, Höst):
  Analyze requirements engineering and management
  for the purpose of validation
  with respect to the state of the practice
  from the point of view of the industrial developers
  in the context of complex systems development.

6

Questions of the Study

- **System development**: general approach, process, artifacts and their relations, tools, logical subsystem architecture
- **Modeling**: notations, model-based RE, methods, DSLs, guidelines, roles, standard terminology
- **Architecture**: responsibility, criteria used, templates, guidelines, criteria and weights
- **Subcontractor relationships**: in-house vs. external, coordination, decision criteria, influence on architecture, documentation received (blackbox?), feature interaction, communication
- **Reuse**: content, extent, guidelines, with subcontractors

7

Hypotheses

- For RE specifications, the main demand by the companies is to adhere to certain document structures.
- A rather low degree of logical modeling is performed during software development.
- The decomposition criteria (the draft version of the criteria catalogue) are rated differently, but tendencies become visible.

8

Results

- There is a defined software development process in every company.
- RE specifications are mainly text-based, sometimes UML diagrams are used.
- The tooling is diverse, with products from, inter alia, Microsoft, Telelogic, and Vector, as well as in-house developed tools.
- The rationale, for example for the decomposition of the system, is usually not documented at all.
- A logical modeling of the system is often skipped for early modeling of the technical architecture.
- Influences from the OEM-subcontractor relationships exist in both directions and efficient communication of requirements and constraints is a challenge.

9

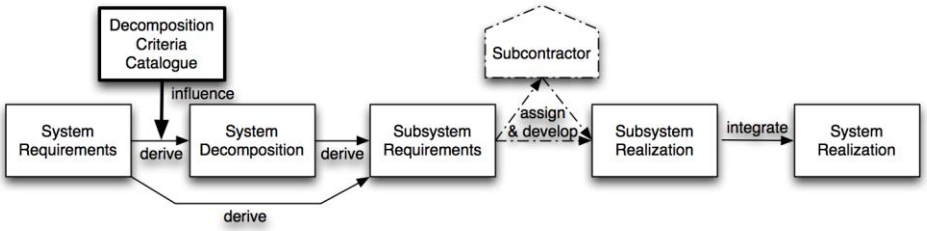
Results for Decomposition Criteria

Table 2.1: Table of Decomposition Criteria and Assigned Weights.

Functional criteria	Logical clustering according to usage	8
	Dependencies	11
	Interaction	10
Architectural criteria	Communication requirements	15
	Technical constraints	12
	Design rules	9
Directive criteria	Laws and standards	10
	Patents, licenses, certificates	8
	Business rules, information politics	4
	Implications from subcontractor relationships	10
Quality criteria	Performance	14
	Correctness, robustness, reliability	14
	Usability	8
	Maintainability	12
	Security	12
	Costs	15

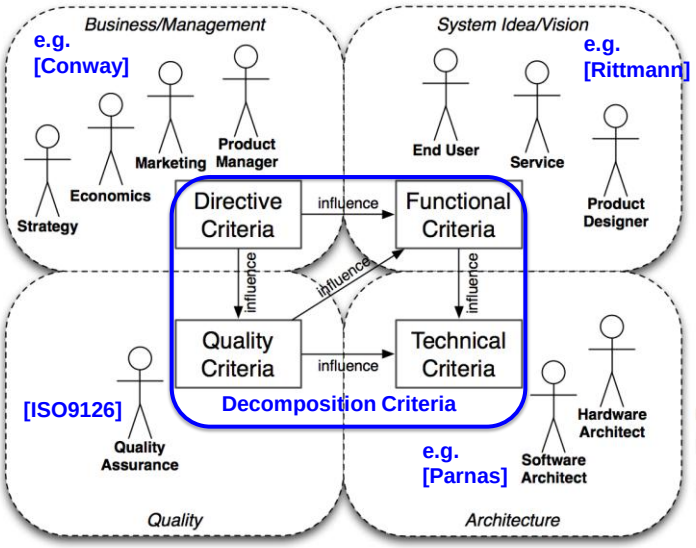
10

DeSyRe Process Overview



11

System Decomposition Criteria



- Use as Checkliste
  - Constructive
  - Analytical
- Evaluation: Interview-based Study

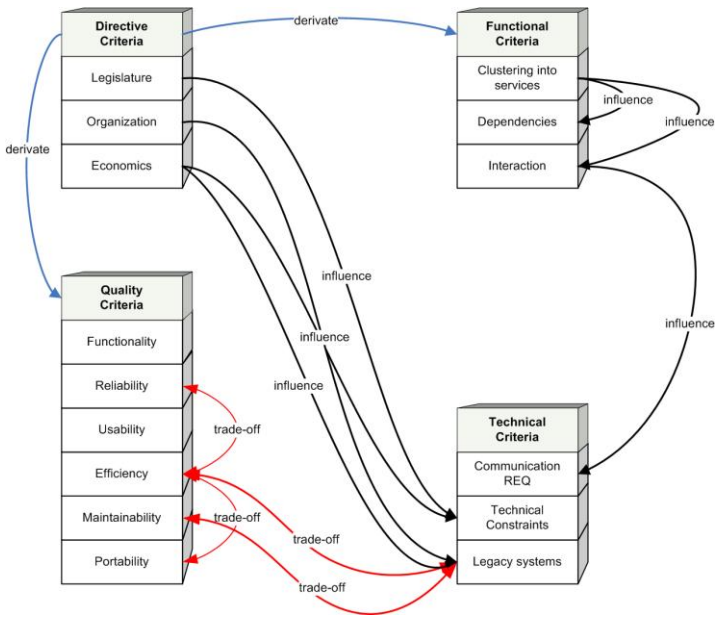
12

Description Template

Template entity	*Description of the entity, to be filled in for each criterion*
Source	<Corresponding documents for information retrieval>
Impact	<Priorities, consequences and risks>
Usage	<Recommendation of state-of-the-art methods>
Examples	<From case study scenario "international logistics company">
Prioritization	<According to the reasons for decomposition, according to the business domain, and according to the system type>

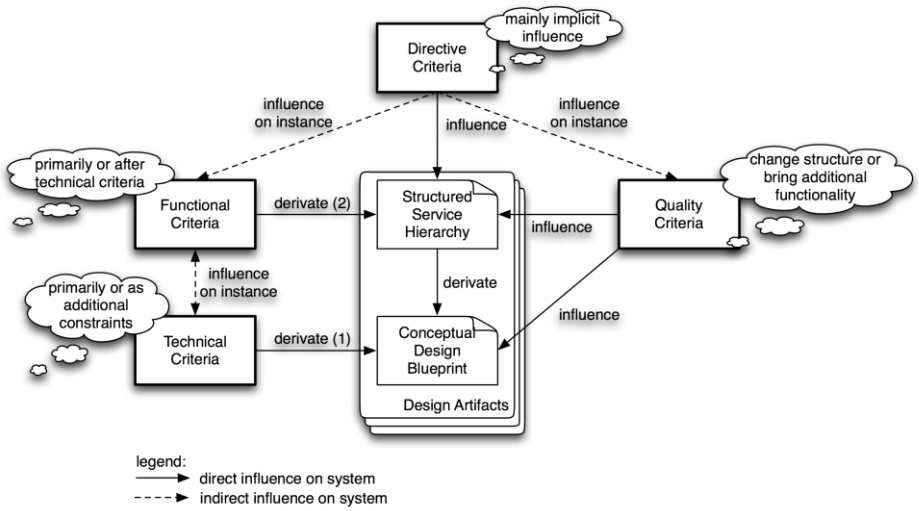
13

Influences between Criteria



14

Application of Catalogue



Example: Decomposition in the Automotive Domain

- 1. The OEM specifies the complete system down to the technical architecture and assigns complete ECUs with the software to be deployed on that ECU to subcontractors.
- 2. The OEM divides the system into hardware and software and signs up different subcontractors for them.
- 3. The OEM decomposes the system according to functionality and assigns functional modules as performed in the aircraft domain, e.g. by Airbus2.
- 4. The OEM distributes a usage function over various ECUs to save resources.

16

Example: Decomposition of Driver Assistance Systems, Functional Feat.

Table 3.6: Clustering according to Functional Features

Source	Scenarios, functional requirements
Impact	"Intuitive" and process-oriented, but no explicit account for quality requirements
Usage	Broy [BKM07], QUASAR enterprise [Sie02], SOA [BS06], Rittmann [Rit08b]
Examples	RFW Feature description: "The system stores information about a possible speed limit during a parking stop and presents it to the driver at system startup." [Ris07, RFW_SL-66] Functional requirement from Diagnostic Service Lane Change Warning: "The system warns the driver about risky lane changes and supports him / her during the execution with a probably necessary correcting reaction to avoid an impending collision." [Gyö08]
Prioritization	According to reason for decomposition (= optimization factors). There is a difference in modelling services for distributed development and for distributed delivery, as the latter has to include considerations about the future usage domains of the system.

17

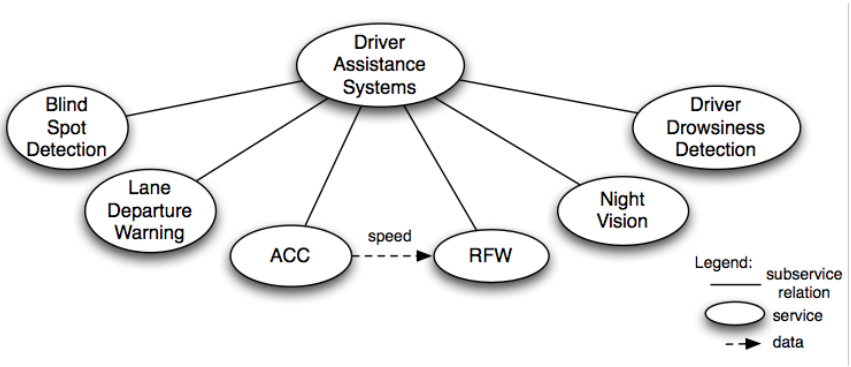
Example: Decomposition of Driver Assistance Systems, Comm. Req.

Table 3.12: Communication Requirements

Source	Behavioural requirements, standards (for system apects, not documented within system specification)
Impact	Transaction security, data consistency, technical adequacy
Usage	Consider for technical architecture, for example, on the SOA layer, see Sensoria [FLB06]
Examples	"The car is not equipped with RFID transmitters. Communication will only flow in one direction, from the mobile or fixed radio signals to the cars." [Ris07, RFW_SL-72] Information system: "The IP protocol has to be used."
Prioritization	According to system type, e.g. for a BIS we think about communication protocols in terms of messages, while for an embedded system we think about the bit patterns for certain sensor values.

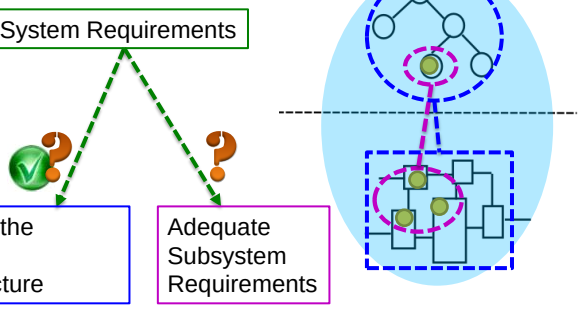
18

Example: Decomposition of Driver Assistance Systems



19

Overview



Challenges:

Draft of the System Architecture

Adequate Subsystem Requirements

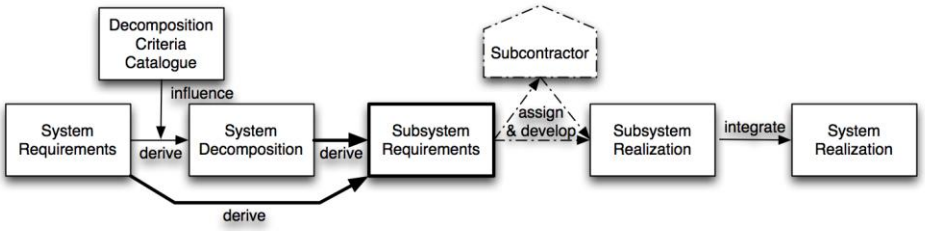
Contributions:

- Catalogue for Decomposition Criteria
- Interview Study on System Decomposition

- Patterns for Requirements Decomposition
- Method for Implementation
- Case Studies on Applicability and Usefulness

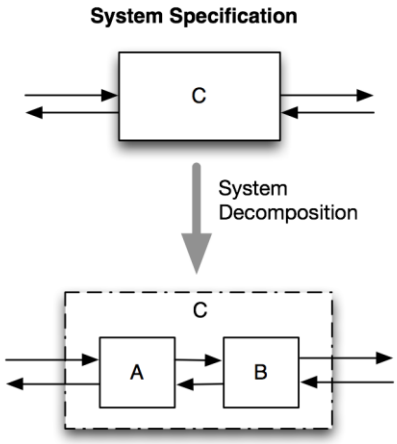
20

DeSyRe Process Overview



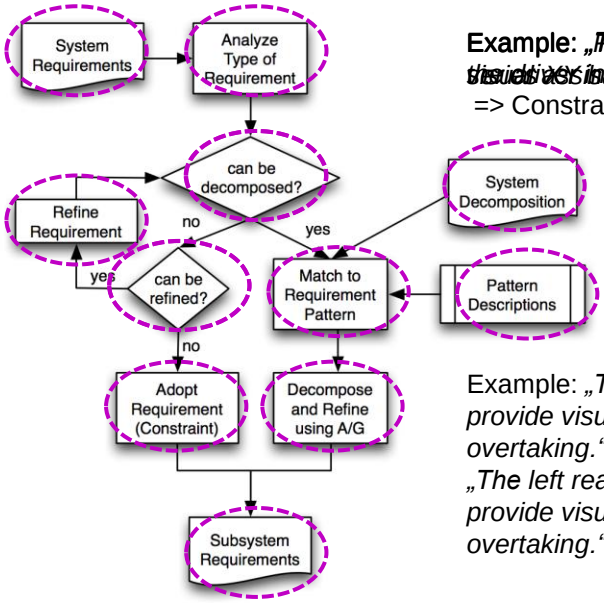
21

Transition from System Requirements to Subsystem Requirements



22

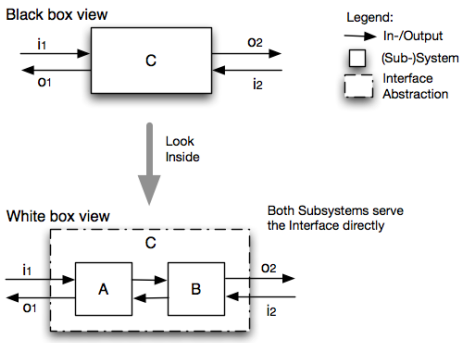
Steps for Refinement and Decomposition



Example: *„The DAS shall provide visual assistance during overtaking.“*
=> Constraint

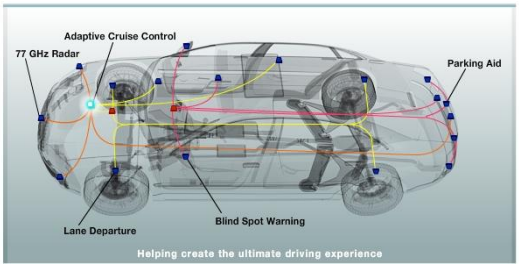
Example: *„The console display shall provide visual assistance during overtaking.“*
„The left rear view mirror shall provide visual assistance during overtaking.“

Decomposition Pattern



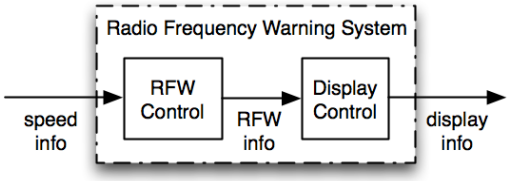
Example: Driver Assistance Systems

- Radio Frequency Warner
- Navigation System
- Adaptive Cruise Control



In Detail: Pattern „Pipeline“

- Requirement: *„In case of speeding, the driver shall be warned by display.“*

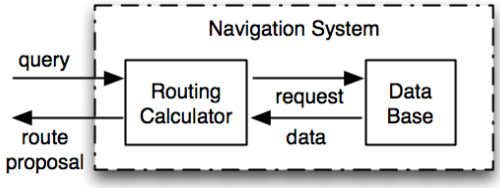


$$A(input, output) \Rightarrow G(input, output)$$

- A(*speedinfo, displayinfo*): Information on permitted speed is available.
- A(*speedinfo, RFWinfo*): Information on permitted velocity is available.
- G(*speedinfo, RFWinfo*): Information sent whether driver is speeding.
- A(*RFWinfo, displayinfo*): Information available on whether driver is speeding.
- G(*RFWinfo, displayinfo*): Driver warned by display when indicated.
- G(*speedinfo, displayinfo*): Driver warned by display when indicated.
- Guarantee of *Controller* fulfills Assumption of *Display*.

26

In Detail: Pattern Subservice

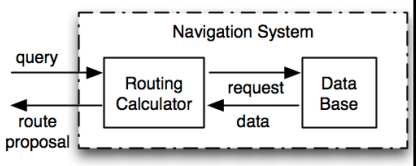


- "The system proposes a route from the point of departure to the chosen destination."
- A (query): There are valid inputs for the point of departure and the destination.
- G (query, route proposal): The system proposes an appropriate route according to the query.

27

In Detail: Pattern Subservice

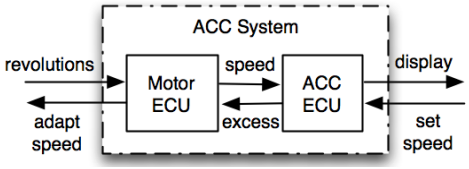
- Subsystem *Routing Calculator (RC)*
  - $A_{RC}(query)$:
    There are valid inputs for the point of departure and the destination.
  - $A_{RC}(data)$:
    The data base delivers correct information.
  - $G_{RC}(query,request)$:
    There are valid inputs for the request.
  - $G_{RC}(data,route\ proposal)$:
    The system proposes an appropriate route.
  - $A_{RC}(query) \wedge A_{RC}(data) \Rightarrow G_{RC}(query,request) \wedge G_{RC}(data,route\ proposal)$



- Subsystem *Data Base (DB)*
  - $A_{DB}(request)$:
    There are valid inputs for the request.
  - $G_{DB}(request,data)$:
    The system delivers correct data about the possible routes between the requested points.
  - $A_{DB}(request) \Rightarrow G_{DB}(request,data)$

28

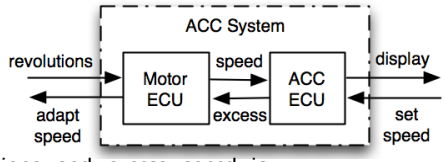
In Detail: General Pattern



- The ACC-System allows the driver to set a specific speed that the vehicle automatically maintains.
- A(revolutions,setspeed):
  The input revolutions delivers the current number of revolutions from the wheels and the input set speed delivers the speed request by the driver.
- G(revolutions,setspeed,display,adaptspeed):
  The output display delivers feedback for the driver according to the request set speed and the output adapt speed sends commands to the motor for adapting the speed.

29

In Detail: General Pattern



- Subsystem *Motor ECU*
  - A_{Motor}(*revolutions*,*excess*):
    The information about current *revolutions* and *excess* speed is available.
  - G_{Motor}(*revolutions*,*excess*,*speed*,*adaptspeed*):
    The current *speed* is calculated from the *revolutions* and the *excess* information is checked whether it is necessary to *adapt speed* is provided.
- Subsystem *ACC ECU*
  - A_{ACC}(*speed*,*setspeed*):
    The information about the current *speed* is available and the input *set speed* delivers the speed request by the driver.
  - G_{ACC}(*speed*,*setspeed*,*excess*,*display*):
    The information about *excess speed* is delivered after comparing the current *speed* to the *set speed* and the feedback is delivered to the driver via *display*.

30

Deduction of nonfunctional Requirements

1. Alternative: Decompose according to System Attributes
   - Example: *„Display texts that shall be read on a display during driving must not be smaller than 1cm.“*

2. Alternative: Decompose with Calculation Model
   - Example: *„The response time of the system shall be less than half a second for 90% system uptime.“*
   - Calculation models necessary, e.g., for probabilities, geometric characteristics, correction algorithms
   - Calculation partially unknown

3. Alternative: Constraint-Handling instead of Decomposition
   - Example: *„The software may not make use of null pointers.“*

31

Deduction of nonfunctional Requirements

Using the classification by Robertson and Robertson 2007 a general guideline for the alternatives is:

1. Alternative: Decompose according to System Attributes
   - look and feel requirements, cultural and political requirements, and many usability requirements.

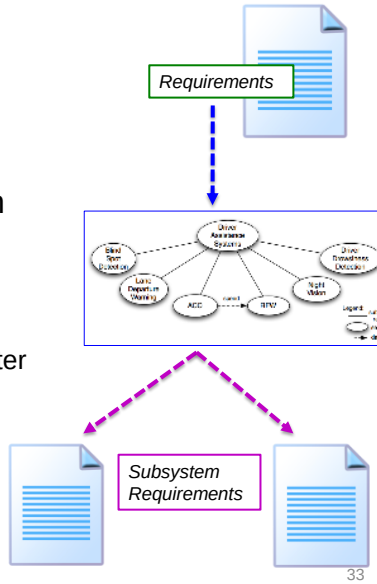
2. Alternative: Decompose with Calculation Model
   - performance requirements, security requirements, some usability requirements, and some legal requirements

3. Alternative: Constraint-Handling instead of Decomposition
   - operational and environmental requirements as well as maintainability and support requirements, and many legal requirements.

32

Evaluation of Applicability

- Case Study on Driver Assistance Systems
- Application of Decomposition criteria catalogue and Requirements Decomposition
- Results:
  - Both applicable.
  - Many seemingly non-decomposable requirements were decomposable after refinement.
  - Higher effort, but deduction continuously traceable.



33

Evaluation of Usefulness

- Presentation in software development company for embedded systems and BIS
- Questionnaire for software engineers

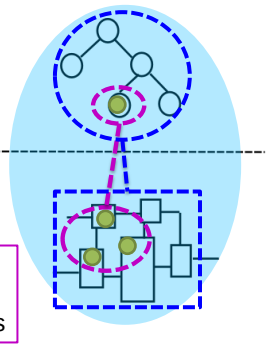
[Davis „Perceived Usefulness“ 1989]

- Approach evaluated positively with respect to improvement of structuredness, completeness, traceability, integration and reusability of requirements.

Questionnaire on usefulness of DeSyRe	Fully agree	Agree	Somewhat agree	Somewhat disagree	Disagree	Fully disagree
Improved structuredness	1	9	1			
Improved completeness	1	5	2	2		
Improved traceability	2	4	2	2		
Easier integration	3	1	4	2	1	
Improved reusability		3	4	2		

34

Overview – Results



System Requirements

Challenges:

Draft of the System Architecture

Adequate Subsystem Requirements

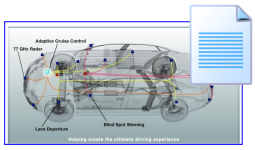
Contributions:

- Catalogue for Decomposition Criteria
- Interview Study on System Decomposition

- Patterns for Requirements Decomposition
- Method for Implementation
- Case Studies on Applicability and Usefulness

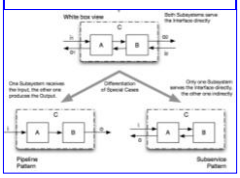
35

Wrapup: Decomposition of Systems and their Requirements

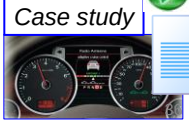


Decomposition Criteria Catalogue

Decomposition Patterns& Method

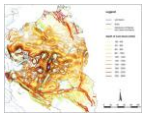
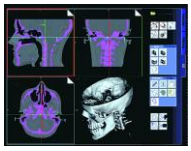


Case study

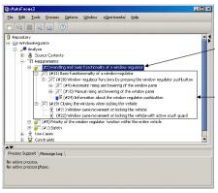


Future Work

- Case study in different application domain


- Tool support in different degrees of automation




Thank you!

mailto:penzenst@in.tum.de

37