Embedded Software Engineering
Part 2: Requirements Basics

Prof. Dr.-Ing. Stefan Kowalewski
Chair "Informatik XI", Embedded Software Laboratory
RWTH Aachen University
kowalewski@informatik.rwth-aachen.de
Summer term 2005

Reminder: V model

requirements analysis → acceptance test

specification → integration test

architecture design → system integration

modul/algorithm design → modul test

implementation
Requirements analysis

- **Objective:**
  - Determine and document the required functionality and properties of the system (What and how good, not how)
  - Determine and document how the achievement of the functionality and properties can be validated or measured, i.e. the test cases for the acceptance test

- **Results:**
  - Specification document
  - Acceptance test plan
  - Possibly first version of system manual

---

Example: Adaptive Cruise Control (ACC)
Context: Car Periphery Supervision

77 GHz Long Range Radar
Fernbereich ≤ 150 m

Infrarot
Nachtsichtbereich ≤ 120 m

Video
Mittelbereich ≤ 80 m

Ultraschall
Ultranahbereich ≤ 3 m

Video
Heckbereich

Graphics: Bosch
Example ACC: Functionality

Example ACC: User Interface

Graphics: DaimlerChrysler
Example ACC: Structure

Fig. 1
1. ACC sensor & control unit
2. Engine management ECU
3. Adaptive intervention in braking via ESP
4. Controls and display
5. Control unit for engine via EDC (Electronic Transmission Control or ETC)
6. Sensors
7. Transmission intervention (optional)

Graphics: Bosch
Example ACC: Sensor

![Example ACC: Sensor](image1)

Example ACC: Sensor /2

![Example ACC: Sensor /2](image2)
**Requirements for ACC /1**

- ACC shall detect vehicles in front of the car in a distance from 0 to 120 meters and in an angle of +/- 4° relative to the centerline of the car.
- ACC shall determine the velocity of detected vehicles.
- If the road is free, ACC will keep the car at a specified cruise velocity.
- If a slower vehicle is in front of the car, ACC will reduce speed until a specified distance to the front vehicle is achieved. ACC will then keep the distance constant until the car either vanishes or accelerates above the specified cruise velocity.
- ACC can be activated between 30 and 180 km/h.
Requirements for ACC /2

- In distance control mode, deviations from the reference distance must not exceed 5% for more than 2 seconds assuming a maximal acceleration or deceleration of the front vehicle of 0.5 g.
- The distance control algorithm shall be a PI controller.
- If ACC is malfunctioning, it has to switch off automatically and indicate by a red light and a short beep that it is out of service. It must never indicate a safe distance to the front vehicle when this is not the case.
- ACC must not be out of function for more than 10e-7 hours per year of operation time.
- Distance and cruise speed selections must be well readable for the driver.

Requirements for ACC /3

- Speed and direction angle information are provided by the ESP system via the CAN bus.
- For customer A, the ACC software has to run on a Motorola MPC4200. Customer B wants to integrate the ACC software into his already existing engine control unit.
- Manufacturer A wants a 77 Ghz radar sensor, manufacturer B an infrared sensor, manufacturer C wants to combine radar and video information.
- Marketing plans to offer ACC Stop&Go with an additional 24 GHz short range radar sensor for next year.
**Requirements Engineering**

Requirements Engineering consists of three parts:

- **Requirements Elicitation**
  - Collect the requirements from customers, marketing, system engineers etc.

- **Requirements Analysis**
  - Analyse whether the requirements are actually what the customers, marketing, system engineers etc. want.

- **Requirements Management**
  - Manage changes of the requirements (document, estimate costs, check technical possibility, delegate to developers, charge customer)

**To which phase of the V-model does Requirements Management belong?**

---

**V model**

- requirements analysis
- specification
- architecture design
- modul/algorithm design
- implementation
- modul test
- system integration
- integration test
- acceptance test
What do you need to know about requirements elicitation and analysis?

- There are **techniques** for requirement elicitation:
  - Guided interviews
  - Brainstorming
  - Role plays, etc. (*→* exercises)

- There are **models** for requirement analysis:
  - Use cases
  - Sequence diagrams
  - Data flow diagrams, etc. (*→* later in the lecture)

- For successful elicitation and analysis of requirements, it is most important to know a few **basic concepts** and **typical problems**.

Requirements vs. solutions

- A requirement should state **what** the system shall do or **how good** it shall do it, not **how**.

- Often, customers tend to provide **solution** aspects with the requirements.

- Example:
  
  **The detected objects shall be stored in a linked list.**

- Often the customer is not aware that he/she demands a particular solution and agrees on replacing it by a proper requirement.

- Example:

  **The detected objects have to be stored such that they can be processed efficiently.**
Requirements vs. constraints

- If a demanded solution cannot be replaced by a requirement, it is called a constraint.
- Example: The system must use Microsoft Windows CE as operating system.
- Behind every constraint there is a rationale which cannot be questioned by the developer.
- Example: The management of the company signed a strategic partnership with Microsoft.

Functional vs. Quality requirements

- Difference between what the system shall do and how good it shall do it.
- Examples:
  - Function (what):
    • The system must keep the distance to the vehicle in front of the car constant.
  - Quality (how good, also non-functional requirements):
    • The driver must be able to adjust the desired distance without taking the hands from the steering wheel. (Usability)
    • Deviations from the desired distance must not exceed 5%. (Reliability)
    • The distance control algorithm must be easily replaced with a customer-owned algorithm. (Modifiability, Integrability)
Sample Qualities

- Maintainability
  - Modifiability
  - Integrability
- Usability
- Dependability
  - Reliability
  - Availability
  - Safety
  - Security

**Particular for embedded systems:**
- System Cost
- Mounting space
- Power consumption

Closed loop vs. Controller requirements

- Distinguish between requirements for **embedded** system and **embedding** system.
- Particular for embedded software:
  The presumed properties and behaviour of the plant (environment) must be elicited and analysed in the requirements phase!

**ACC example?**
Requirements must be checkable

- Both functional and non-functional requirements must be formulated such that their fulfillment can be checked.
- Examples:
  - **Bad**: The ACC system shall react properly when the car in front becomes too slow.
  - **Good**: If the speed of car in front becomes smaller than 30 km/h, ACC shall stop controlling the distance, release control over brake and accelerator back to the driver, and indicate this situation with a loud beep.
  - **Bad**: The software shall be prepared for a change of the measurement principle (e.g., infrared instead of radar)
  - **Good**: The necessary software adaptations for changing the principle of the sensor must be performable by one developer in one week.

Requirements must understandable by the developers

- In embedded systems:
  - How much **domain knowledge** can be expected from the developers?
- Example:
  - **For the distance control, overshoot shall be below 5% and rise time below 3 seconds.**
- Trade-off between presumed domain knowledge of developer and size/complexity of requirements.
- A **dictionary** is always helpful.
Concise requirements require sufficient
domain knowledge of the developers

- A famous example from a famous requirements analysis
  book (D.C. Gause, G.M. Weinberg: Are your lights on?
  Adapted from Philip Koopman, CMU):
  - Shortly after a road tunnel there is a scenic-view overlook
    with a parking area.
  - Before the tunnel there is a sign asking the car drivers to
    switch on the lights.
  - Result: Many drivers go through the tunnel, turn into the
    parking area, forget to switch off the lights, and have flat
    batteries when they return.
  - Therefore the highway department intends to erect a sign
    after the tunnel which specifies the appropriate behavior
    with respect to switching off the lights.

First try

Turn your lights off

- But what about those who did not switch their lights on in
  the tunnel?
Second try

If your headlights are on, turn them off

- But what if it is night time?

Third try

If it is daytime and your headlights are on, turn them off

- But what if there is fog and visibility is reduced during daytime?
Fourth try

If your headlights are on, and they are not required for visibility, turn them off

- But what about modern (US) cars which are designed such that the headlights are on whenever the motor is running?

Fifth try

If your headlights are on, and they are not required for visibility, and you can turn them off, then turn them off

What would you do?
Lessons from this example?

Summary: Basic requirements issues

Basic concepts and typical problems during requirements elicitation:
- Solutions instead of requirements
- Difference between solutions and constraints
- Difference between functional and non-functional (quality) requirements
- Closed loop vs. Controller requirements
- Assumed environment (plant) behavior and properties must be elicitated
- Requirements must be checkable
- Requirements must be understandable
### Requirements for ACC /1

- ACC shall detect vehicles in front of the car in a distance from 0 to 120 meters and in an angle of +/- 4° relativ to the centerline of the car.
- ACC shall determine the velocity of detected vehicles
- If the road is free, ACC will keep the car at a specified cruise velocity.
- If a slower vehicle is in front of the car, ACC will reduce speed until a specified distance to the front vehicle is achieved. ACC will then keep the distance constant until the car either vanishes or accelerates above the specified cruise velocity.
- ACC can be activated between 30 and 180 km/h.

### Requirements for ACC /2

- In distance control mode, deviations from the reference distance must not exceed 5% for more than 2 seconds assuming a maximal acceleration or deceleration of the front vehicle of 0.5 g.
- The distance control algorithm shall be a PI controller.
- If ACC is malfunctioning, it has to switch off automatically and indicate by a red light and a short beep that it is out of service. It must never indicate a safe distance to the front vehicle when this is not the case.
- ACC must not be out of function for more than 10e-7 hours per year of operation time.
- Distance and cruise speed selections must be well readable for the driver.
Requirements for ACC /3

- Speed and direction angle information are provided by the ESP system via the CAN bus.
- For customer A, the ACC software has to run on a Motorola MPC4200. Customer B wants to integrate the ACC software into his already existing engine control unit.
- Manufacturer A wants a 77 Ghz radar sensor, manufacturer B an infrared sensor, manufacturer C wants to combine radar and video information.
- Marketing plans to offer ACC Stop&Go with an additional 24 GHz short range radar sensor for next year.