Modeling Textual Requirements

CPS Course

(based on IT System Design)

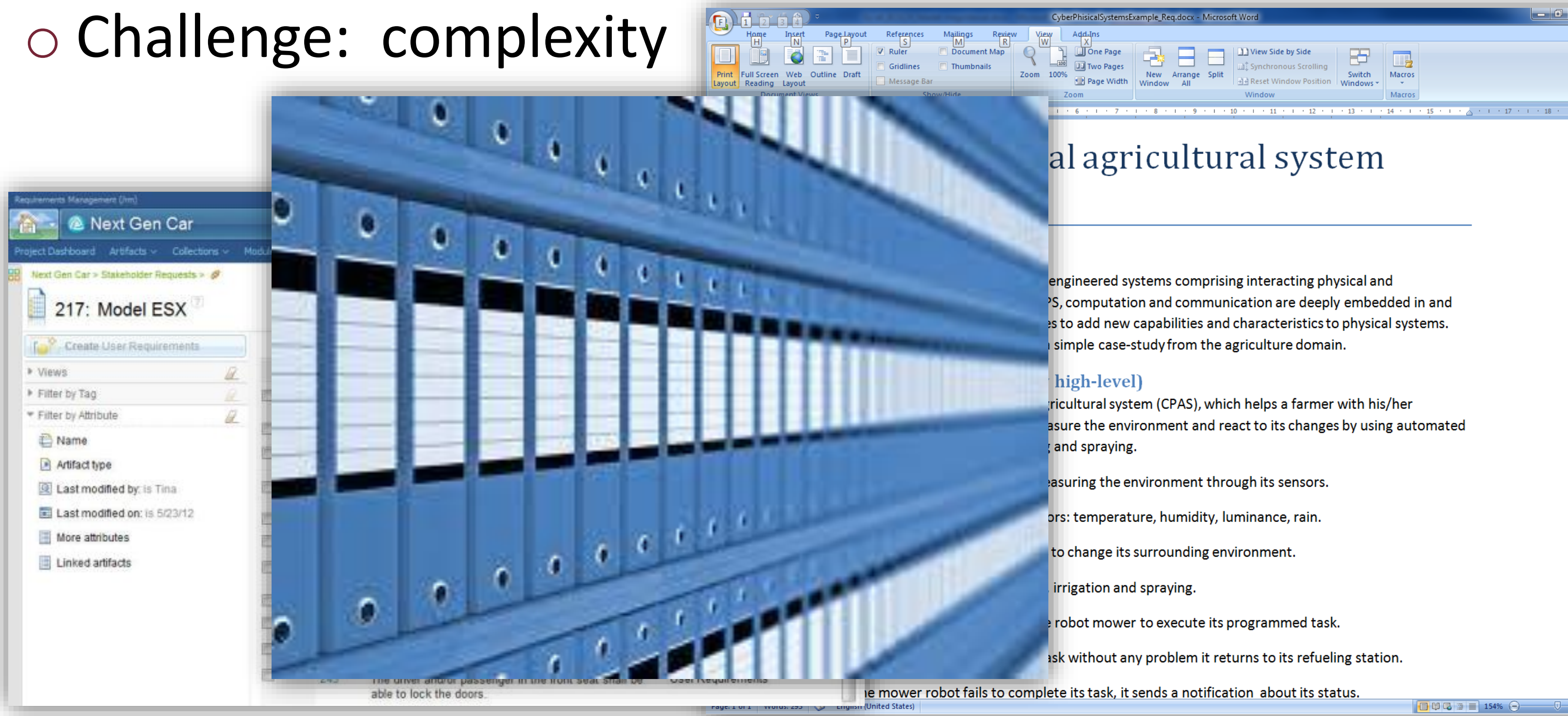
Budapest University of Technology and Economics
Department of Measurement and Information Systems

What is it about?

Context of the Modeling Aspect

Roots & Relations

- Document based system development
  - Formulated requirements textually (e.g. in Word)
  - Handled by Req. management tools (e.g. DOORS)
  - Challenge: complexity



al agricultural system

engineered systems comprising interacting physical and
PS, computation and communication are deeply embedded in and
es to add new capabilities and characteristics to physical systems.
simple case-study from the agriculture domain.

high-level)

ricultural system (CPAS), which helps a farmer with his/her
asure the environment and react to its changes by using automated
and spraying.

asuring the environment through its sensors.

ors: temperature, humidity, luminance, rain.

to change its surrounding environment.

irrigation and spraying.

robot mower to execute its programmed task.

sk without any problem it returns to its refueling station.

e mower robot fails to complete its task, it sends a notification about its status.

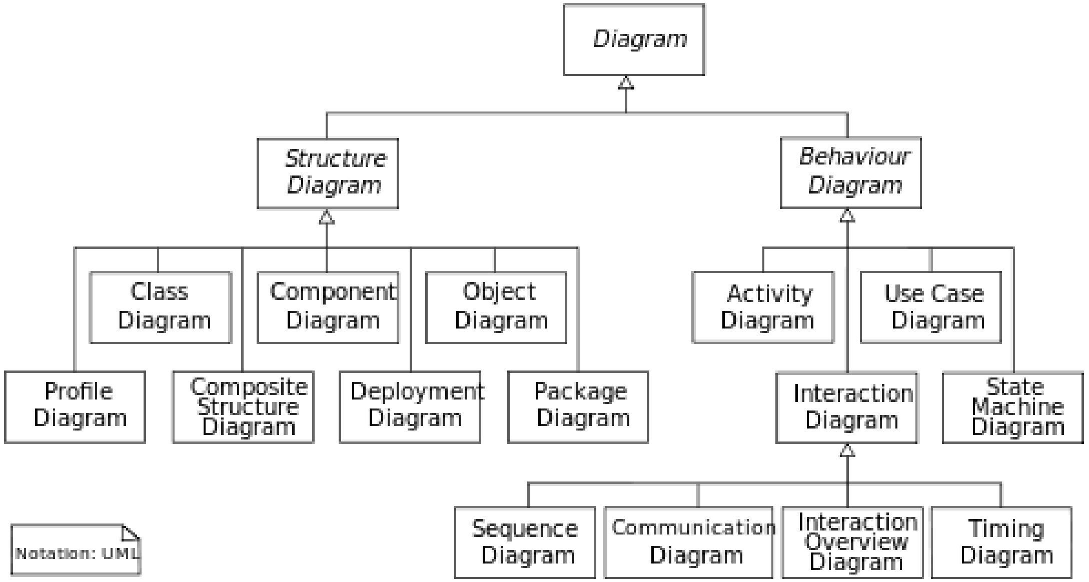
Next Gen Car

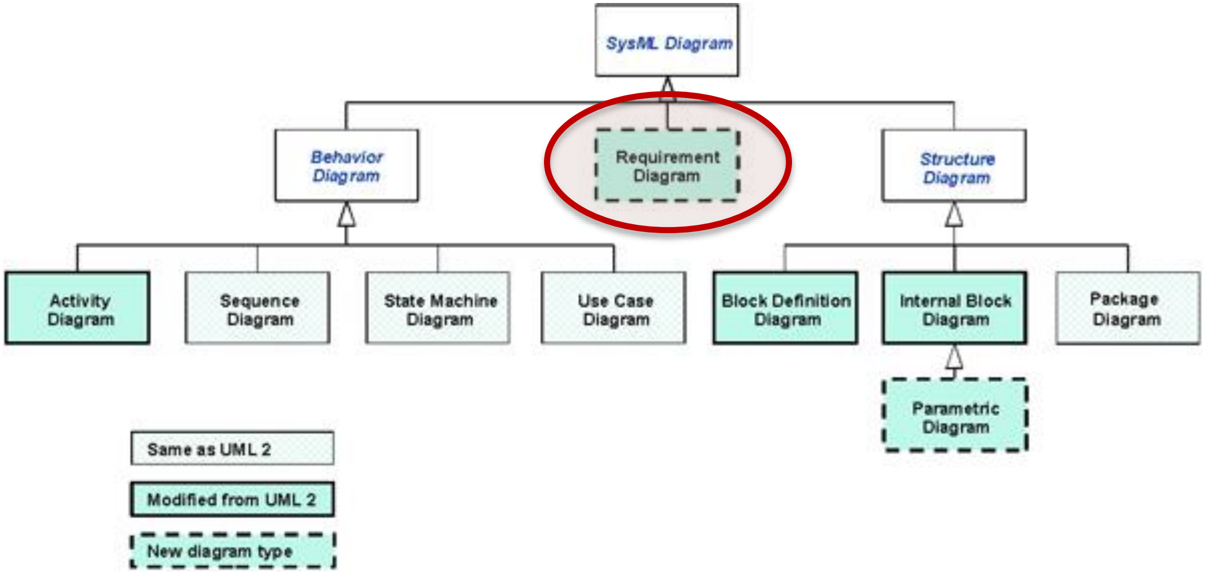
Next Gen Car > Stakeholder Requests >

217: Model ESX

The driver and/or passenger in the front seat shall be able to lock the doors.

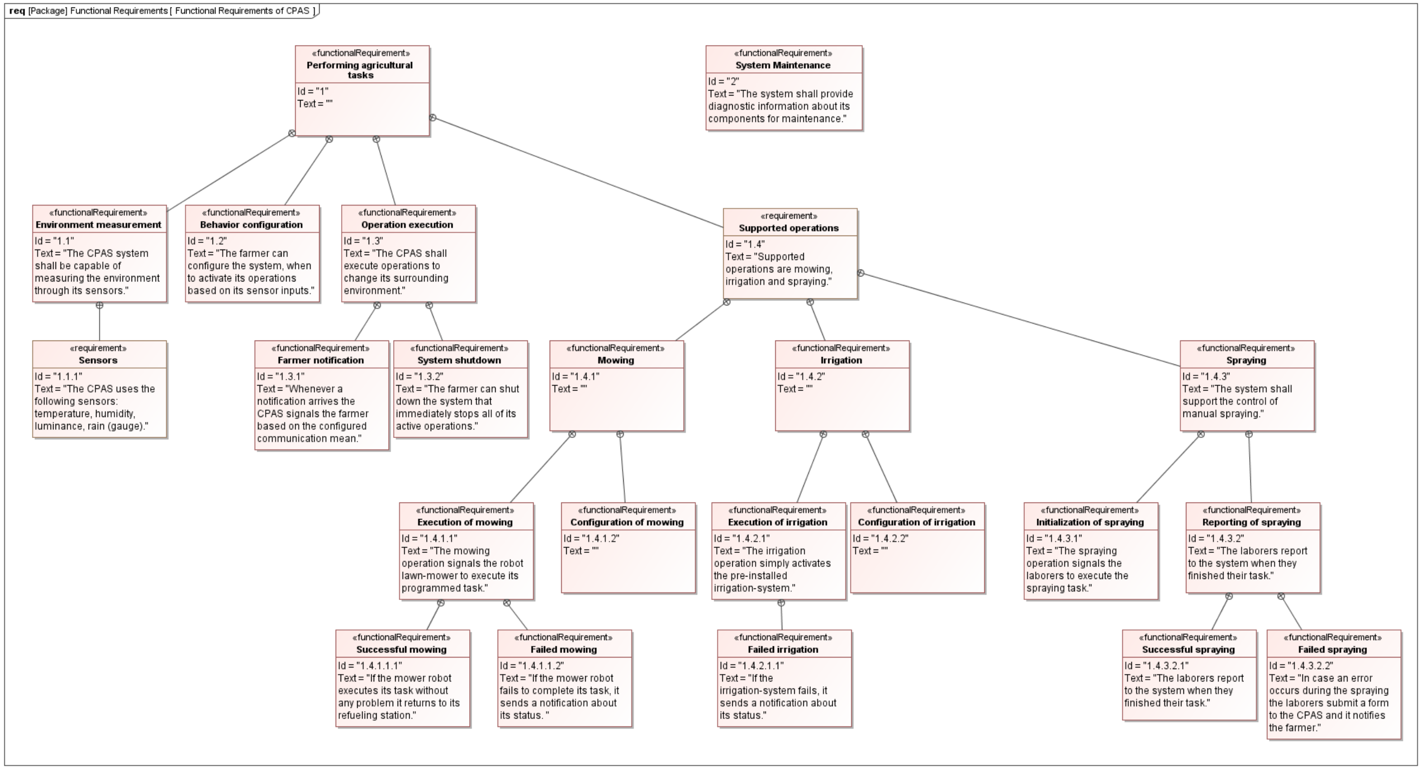
Requirements Diagram





Modeling Aspect

What are the main requirements formulated textually and what are their hierarchy?



req [Package] Functional Requirements [Functional Requirements of CPAS]

«functionalRequirement»
Performing agricultural tasks
Id = "1"
Text = ""

«functionalRequirement»
System Maintenance
Id = "2"
Text = "The system shall provide diagnostic information about its components for maintenance."

«functionalRequirement»
Environment measurement
Id = "1.1"
Text = "The CPAS system shall be capable of measuring the environment through its sensors."

«functionalRequirement»
Behavior configuration
Id = "1.2"
Text = "The farmer can configure the system, when to activate its operations based on its sensor inputs."

«functionalRequirement»
Operation execution
Id = "1.3"
Text = "The CPAS shall execute operations to change its surrounding environment."

«requirement»
Supported operations
Id = "1.4"
Text = "Supported operations are mowing, irrigation and spraying."

«requirement»
Sensors
Id = "1.1.1"
Text = "The CPAS uses the following sensors: temperature, humidity, luminance, rain (gauge)."

«functionalRequirement»
Farmer notification
Id = "1.3.1"
Text = "Whenever a notification arrives the CPAS signals the farmer based on the configured communication mean."

«functionalRequirement»
System shutdown
Id = "1.3.2"
Text = "The farmer can shut down the system that immediately stops all of its active operations."

«functionalRequirement»
Mowing
Id = "1.4.1"
Text = ""

«functionalRequirement»
Irrigation
Id = "1.4.2"
Text = ""

«functionalRequirement»
Spraying
Id = "1.4.3"
Text = "The system shall support the control of manual spraying."

«functionalRequirement»
Execution of mowing
Id = "1.4.1.1"
Text = "The mowing operation signals the robot lawn-mower to execute its programmed task."

«functionalRequirement»
Configuration of mowing
Id = "1.4.1.2"
Text = ""

«functionalRequirement»
Execution of irrigation
Id = "1.4.2.1"
Text = "The irrigation operation simply activates the pre-installed irrigation-system."

«functionalRequirement»
Configuration of irrigation
Id = "1.4.2.2"
Text = ""

«functionalRequirement»
Initialization of spraying
Id = "1.4.3.1"
Text = "The spraying operation signals the laborers to execute the spraying task."

«functionalRequirement»
Reporting of spraying
Id = "1.4.3.2"
Text = "The laborers report to the system when they finished their task."

«functionalRequirement»
Successful mowing
Id = "1.4.1.1.1"
Text = "If the mower robot executes its task without any problem it returns to its refueling station."

«functionalRequirement»
Failed mowing
Id = "1.4.1.1.2"
Text = "If the mower robot fails to complete its task, it sends a notification about its status. "

«functionalRequirement»
Failed irrigation
Id = "1.4.2.1.1"
Text = "If the irrigation-system fails, it sends a notification about its status."

«functionalRequirement»
Successful spraying
Id = "1.4.3.2.1"
Text = "The laborers report to the system when they finished their task."

«functionalRequirement»
Failed spraying
Id = "1.4.3.2.2"
Text = "In case an error occurs during the spraying the laborers submit a form to the CPAS and it notifies the farmer."

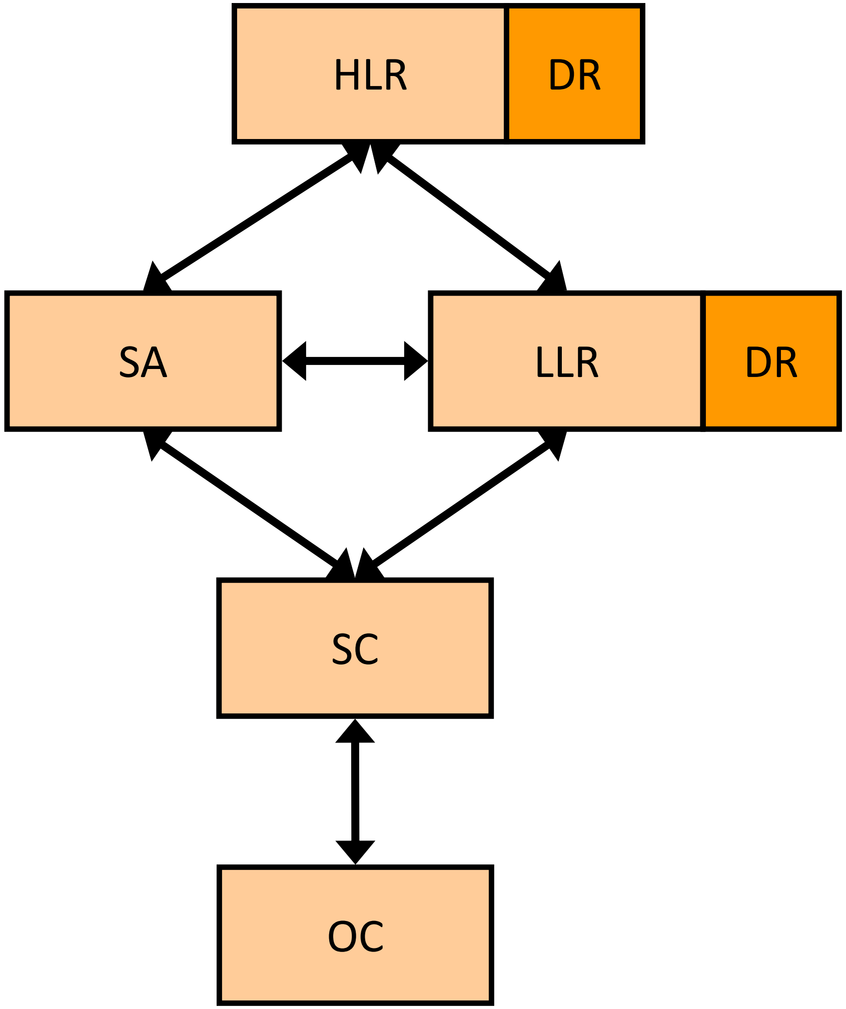
Objectives

- Provides linkage between traditional textual and model based requirements specifications
- Helps establishing relations between requirements
  - Containment hierarchy
  - Derivation
  - Reusing between projects
- Provides traceability of requirements

Definition of a Requirement

- Definitions
  - A condition or capability a system must conform to (IBM Rational)
  - A statement of the functions required of the system (Mentor Graphics)
- Each requirements needs to be
  - **Identifiable + Unique**: unique IDs
  - **Consistent**: no contradiction
  - **Unambiguous**: one interpretation
  - **Verifiable**: e.g. testable to decide if met
- Captured with special statements and vocabulary

The Certification Perspective: High-level vs Low-Level



Concepts from DO-178C standard

- High Level Requirements (HLR):
  - customer-oriented
  - black-box view of the software,
  - captured in a natural language (e.g. using shall statements)
- Derived Requirements (DR)
  - Capture design decisions
- Low Level Requirements (LLR):
  - SC can be implemented without further information
- Software Architecture (SA)
  - Interfaces, information flow of SW components
- Source Code (SC)
- Executable Object Code (EOC)

Functional vs Extra-functional

Functional

- Specific to a component of the system
- Core technical functionality

Extra-functional

- Fulfilled by the system as a whole
- Performance
- Reliability
- Safety
- Security

Sample Requirements

Requirements of a
Cyber-physical Agricultural System

Cyber-physical system

- American terminology
  - Novel buzz-word for embedded system
  - In EU it is ~ „Internet of things”

„ Cyber-Physical Systems (CPS) are engineered systems comprising interacting physical and computational components. In CPS, computation and communication are deeply embedded in and interacting with physical processes to add new capabilities and characteristics to physical systems.”

- E.g., acoustic sniper detection system

Example requirements

Design a simple Cyber-physical agricultural system (CPAS), which helps a farmer with his/her everyday life using sensors to measure the environment and react to its changes by using automated operations like irrigation, mowing and spraying.

Requirements

- The CPAS system is capable of measuring the environment through its sensors.
- The CPAS uses the following sensors: temperature, humidity, luminance, rain.
- The CPAS can execute operations to change its surrounding environment.
- These operations can be mowing, irrigation and spraying.
- The mowing operation signals the robot mower to execute its programmed task.
- If the mower robot executes its task without any problem it returns to its refueling station.
- If the mower robot fails to complete its task, it sends a notification about its status

Example requirements (con’t)

- The irrigation operation simply activates the pre-installed irrigation-system.
- If the irrigation-system fails, it sends a notification about its status.
- Whenever a notification arrives the CPAS signals the farmer based on the configured communication mean.
- The spraying operation signals the laborers to execute the spraying task.
- The laborers report to the CPAS when they finished their task.
- In case an error occurs during the spraying the laborers submit a form to the CPAS and it notifies the farmer.
- The farmer can configure the system, when to activate its operations based on its sensor inputs.
- The farmer can shut down the CPAS system that immediately stops all of its active operations.
- The system shall provide diagnostic information about its components for maintenance.

How to Write Requirement?

Best practices for writing textual requirements

- A textual requirement contains
  - a short description(stand-alone sentence / paragraph)
  - of the problem and not the solution
- English phrasing:
  - Pattern: Subject Auxiliary Verb Object Conditions
    - Example:
      The railway operator shall create a direct route
      between any two points on the track
  - Be precise! (Quantitative is better than qualitative)
  - Avoid passive sentences
- Use of auxiliaries:
  - Positive: shall/must > should > may
  - Negative: must not > may not
  - They specify priorities!

Examples

Functional:

- The operator shall be able to change the direction of turnouts
- Train equipments shall periodically log sensor data with a timestamp

Safety:

- The system shall ensure safe traffic within a zone
- The system shall stop two trains if they are closer than a minimal distance
- No single faults shall result in system failure

Performance:

- The system should allow five trains per every 10 minutes

Reliability:

- The allowed downtime of the system should be less than 1 hour per year
- The system shall continue normal operation within 10 minutes after a failure

Supportability:

- The system shall allow remote access for maintenance

Security:

- The system shall provide remote access only to authorized personnel

Usability:

- The user interface should contain only 3 alerts at a time

Anti-patterns

1. The system should be safe
2. The system shall use Fast Fourier Transformation to calculate signal value.
3. The system shall continue normal operation soon after a failure.
4. Sensor data shall be logged by a timestamp
5. Unauthorized personnel could not access the system

Too general / high-level

Describes a solution (and not only the problem)

Imprecise (how to verify „soon”?)

Passive should be avoided!

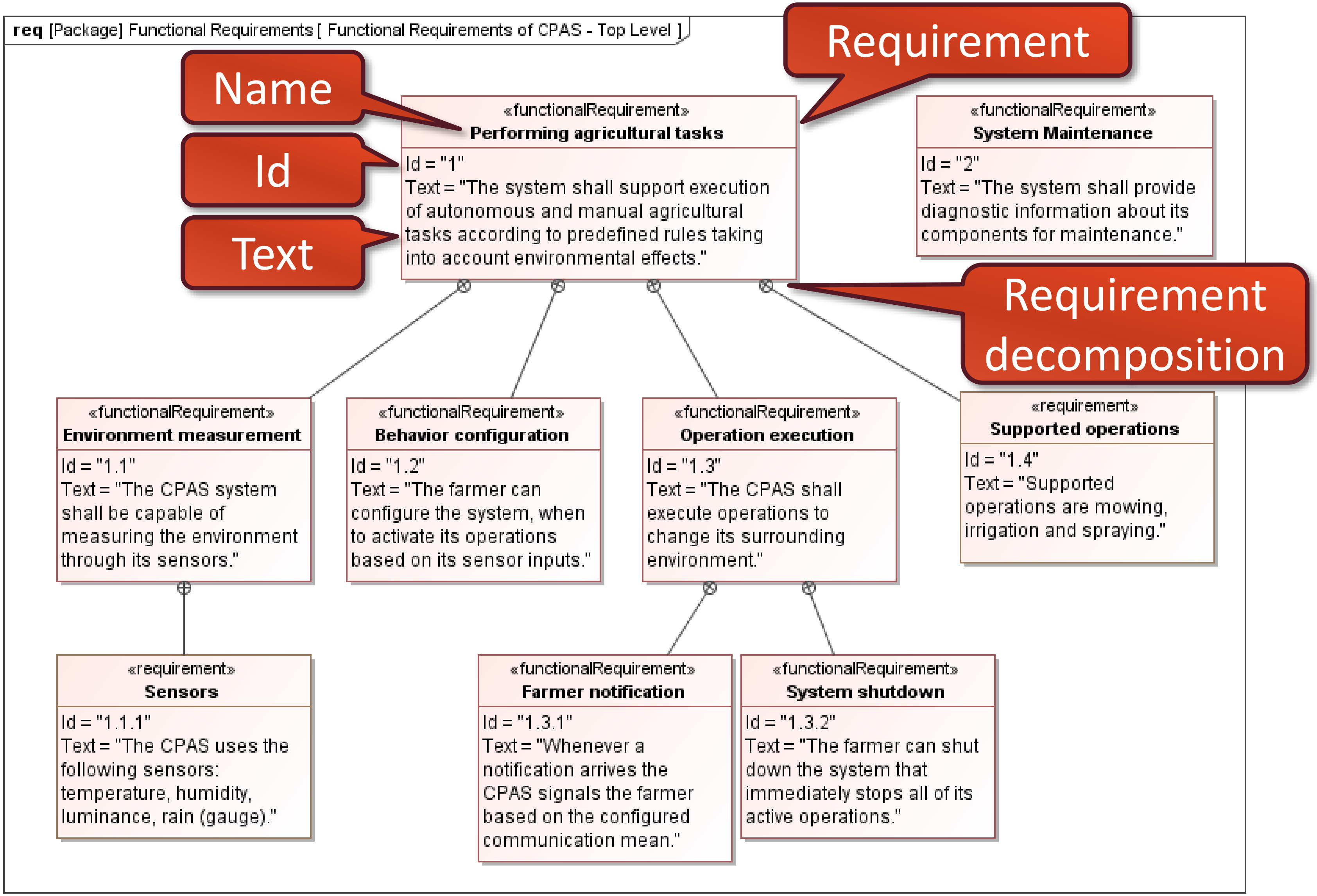
Use specific auxiliaries!

How to identify missing or inconsistent requirements?

What are the building blocks?

Modeling Elements & Notation

Example – Top Level Requirements



req [Package] Functional Requirements [Functional Requirements of CPAS - Top Level]

Name

Id

Text

Requirement

Requirement decomposition

«functionalRequirement»
Performing agricultural tasks
Id = "1"
Text = "The system shall support execution of autonomous and manual agricultural tasks according to predefined rules taking into account environmental effects."

«functionalRequirement»
System Maintenance
Id = "2"
Text = "The system shall provide diagnostic information about its components for maintenance."

«functionalRequirement»
Environment measurement
Id = "1.1"
Text = "The CPAS system shall be capable of measuring the environment through its sensors."

«functionalRequirement»
Behavior configuration
Id = "1.2"
Text = "The farmer can configure the system, when to activate its operations based on its sensor inputs."

«functionalRequirement»
Operation execution
Id = "1.3"
Text = "The CPAS shall execute operations to change its surrounding environment."

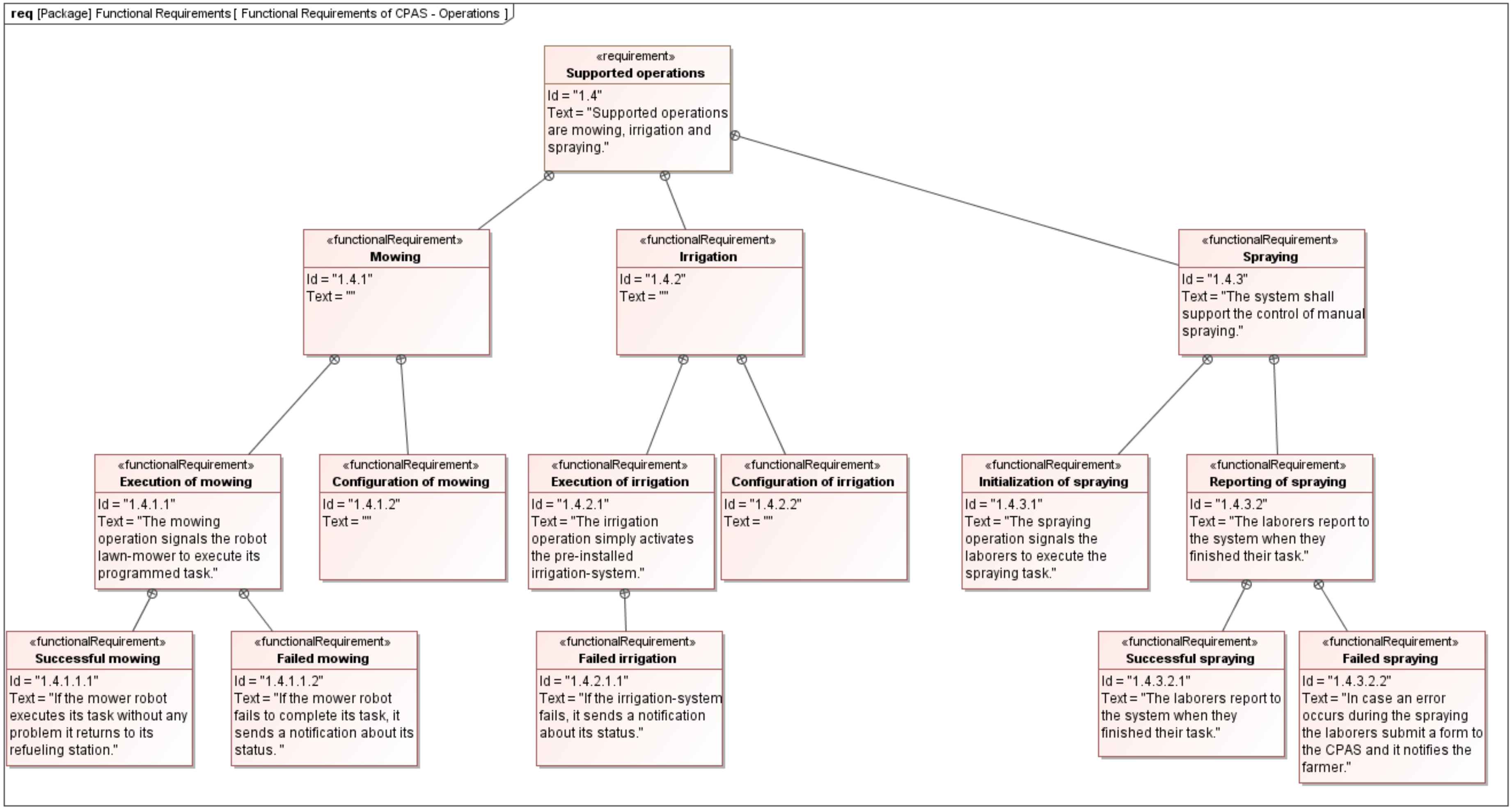
«requirement»
Supported operations
Id = "1.4"
Text = "Supported operations are mowing, irrigation and spraying."

«requirement»
Sensors
Id = "1.1.1"
Text = "The CPAS uses the following sensors: temperature, humidity, luminance, rain (gauge)."

«functionalRequirement»
Farmer notification
Id = "1.3.1"
Text = "Whenever a notification arrives the CPAS signals the farmer based on the configured communication mean."

«functionalRequirement»
System shutdown
Id = "1.3.2"
Text = "The farmer can shut down the system that immediately stops all of its active operations."

Example – Further Decomposed



req [Package] Functional Requirements [Functional Requirements of CPAS - Operations]

«requirement»
Supported operations
Id = "1.4"
Text = "Supported operations are mowing, irrigation and spraying."

«functionalRequirement»
Mowing
Id = "1.4.1"
Text = ""

«functionalRequirement»
Irrigation
Id = "1.4.2"
Text = ""

«functionalRequirement»
Spraying
Id = "1.4.3"
Text = "The system shall support the control of manual spraying."

«functionalRequirement»
Execution of mowing
Id = "1.4.1.1"
Text = "The mowing operation signals the robot lawn-mower to execute its programmed task."

«functionalRequirement»
Configuration of mowing
Id = "1.4.1.2"
Text = ""

«functionalRequirement»
Execution of irrigation
Id = "1.4.2.1"
Text = "The irrigation operation simply activates the pre-installed irrigation-system."

«functionalRequirement»
Configuration of irrigation
Id = "1.4.2.2"
Text = ""

«functionalRequirement»
Initialization of spraying
Id = "1.4.3.1"
Text = "The spraying operation signals the laborers to execute the spraying task."

«functionalRequirement»
Reporting of spraying
Id = "1.4.3.2"
Text = "The laborers report to the system when they finished their task."

«functionalRequirement»
Successful mowing
Id = "1.4.1.1.1"
Text = "If the mower robot executes its task without any problem it returns to its refueling station."

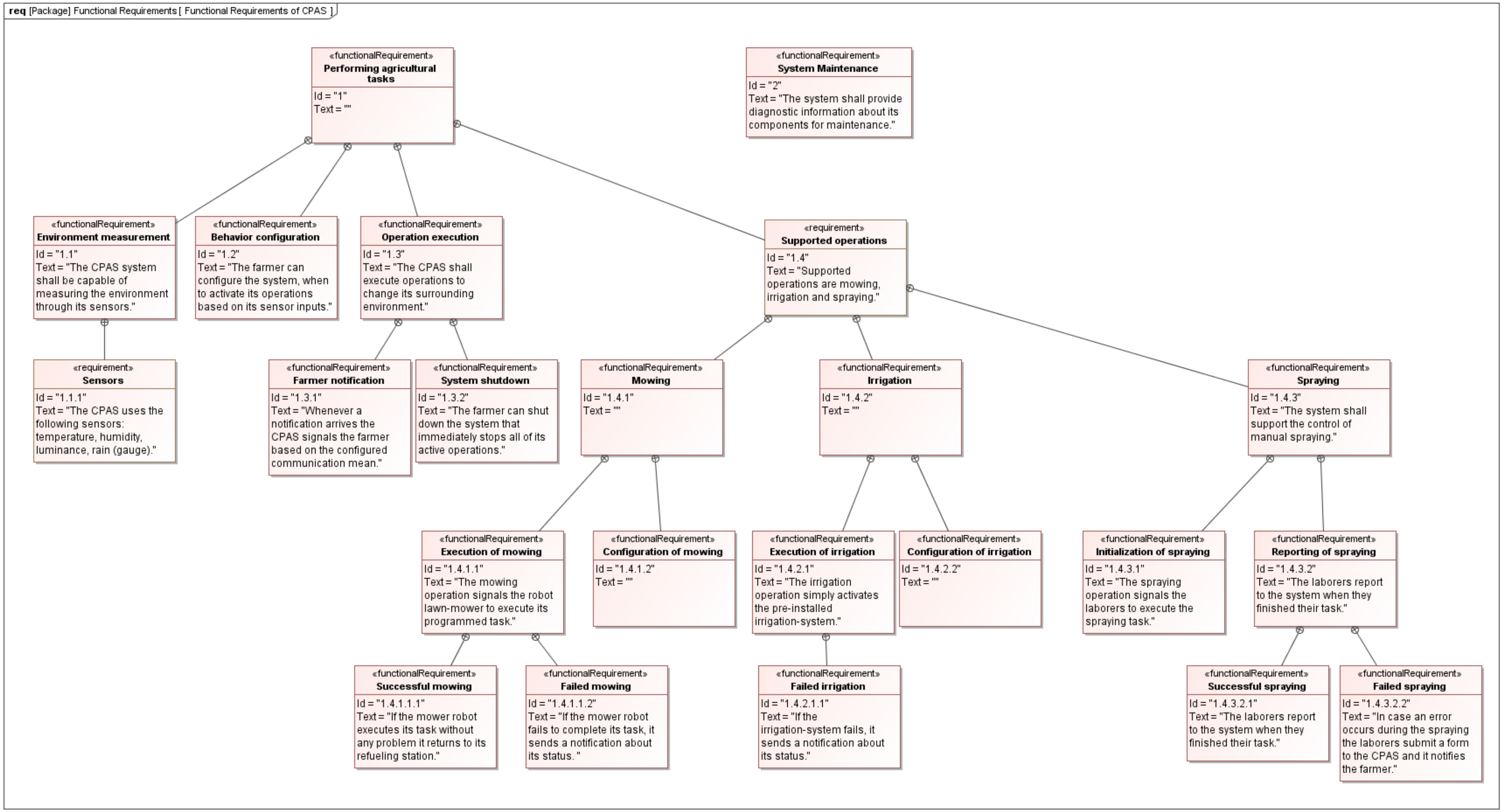
«functionalRequirement»
Failed mowing
Id = "1.4.1.1.2"
Text = "If the mower robot fails to complete its task, it sends a notification about its status. "

«functionalRequirement»
Failed irrigation
Id = "1.4.2.1.1"
Text = "If the irrigation-system fails, it sends a notification about its status."

«functionalRequirement»
Successful spraying
Id = "1.4.3.2.1"
Text = "The laborers report to the system when they finished their task."

«functionalRequirement»
Failed spraying
Id = "1.4.3.2.2"
Text = "In case an error occurs during the spraying the laborers submit a form to the CPAS and it notifies the farmer."

Example – Full Hierarchy



req [Package] Functional Requirements [Functional Requirements of CPAS]

«functionalRequirement»
Performing agricultural tasks
Id = "1"
Text = ""

«functionalRequirement»
System Maintenance
Id = "2"
Text = "The system shall provide diagnostic information about its components for maintenance."

«functionalRequirement»
Environment measurement
Id = "1.1"
Text = "The CPAS system shall be capable of measuring the environment through its sensors."

«functionalRequirement»
Behavior configuration
Id = "1.2"
Text = "The farmer can configure the system, when to activate its operations based on its sensor inputs."

«functionalRequirement»
Operation execution
Id = "1.3"
Text = "The CPAS shall execute operations to change its surrounding environment."

«requirement»
Supported operations
Id = "1.4"
Text = "Supported operations are mowing, irrigation and spraying."

«requirement»
Sensors
Id = "1.1.1"
Text = "The CPAS uses the following sensors: temperature, humidity, luminance, rain (gauge)."

«functionalRequirement»
Farmer notification
Id = "1.3.1"
Text = "Whenever a notification arrives the CPAS signals the farmer based on the configured communication mean."

«functionalRequirement»
System shutdown
Id = "1.3.2"
Text = "The farmer can shut down the system that immediately stops all of its active operations."

«functionalRequirement»
Mowing
Id = "1.4.1"
Text = ""

«functionalRequirement»
Irrigation
Id = "1.4.2"
Text = ""

«functionalRequirement»
Spraying
Id = "1.4.3"
Text = "The system shall support the control of manual spraying."

«functionalRequirement»
Execution of mowing
Id = "1.4.1.1"
Text = "The mowing operation signals the robot lawn-mower to execute its programmed task."

«functionalRequirement»
Configuration of mowing
Id = "1.4.1.2"
Text = ""

«functionalRequirement»
Execution of irrigation
Id = "1.4.2.1"
Text = "The irrigation operation simply activates the pre-installed irrigation-system."

«functionalRequirement»
Configuration of irrigation
Id = "1.4.2.2"
Text = ""

«functionalRequirement»
Initialization of spraying
Id = "1.4.3.1"
Text = "The spraying operation signals the laborers to execute the spraying task."

«functionalRequirement»
Reporting of spraying
Id = "1.4.3.2"
Text = "The laborers report to the system when they finished their task."

«functionalRequirement»
Successful mowing
Id = "1.4.1.1.1"
Text = "If the mower robot executes its task without any problem it returns to its refueling station."

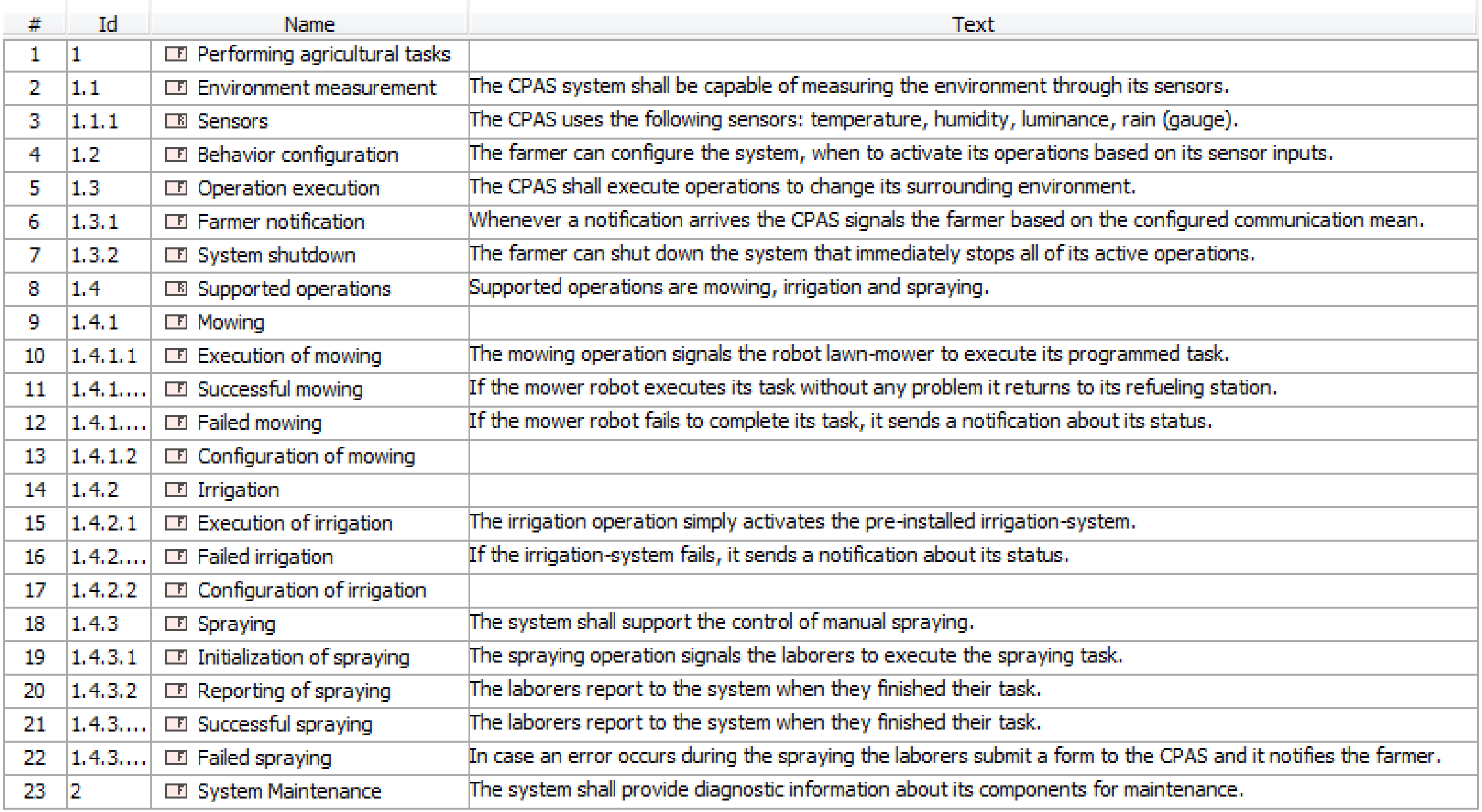
«functionalRequirement»
Failed mowing
Id = "1.4.1.1.2"
Text = "If the mower robot fails to complete its task, it sends a notification about its status. "

«functionalRequirement»
Failed irrigation
Id = "1.4.2.1.1"
Text = "If the irrigation-system fails, it sends a notification about its status."

«functionalRequirement»
Successful spraying
Id = "1.4.3.2.1"
Text = "The laborers report to the system when they finished their task."

«functionalRequirement»
Failed spraying
Id = "1.4.3.2.2"
Text = "In case an error occurs during the spraying the laborers submit a form to the CPAS and it notifies the farmer."

Requirements Table

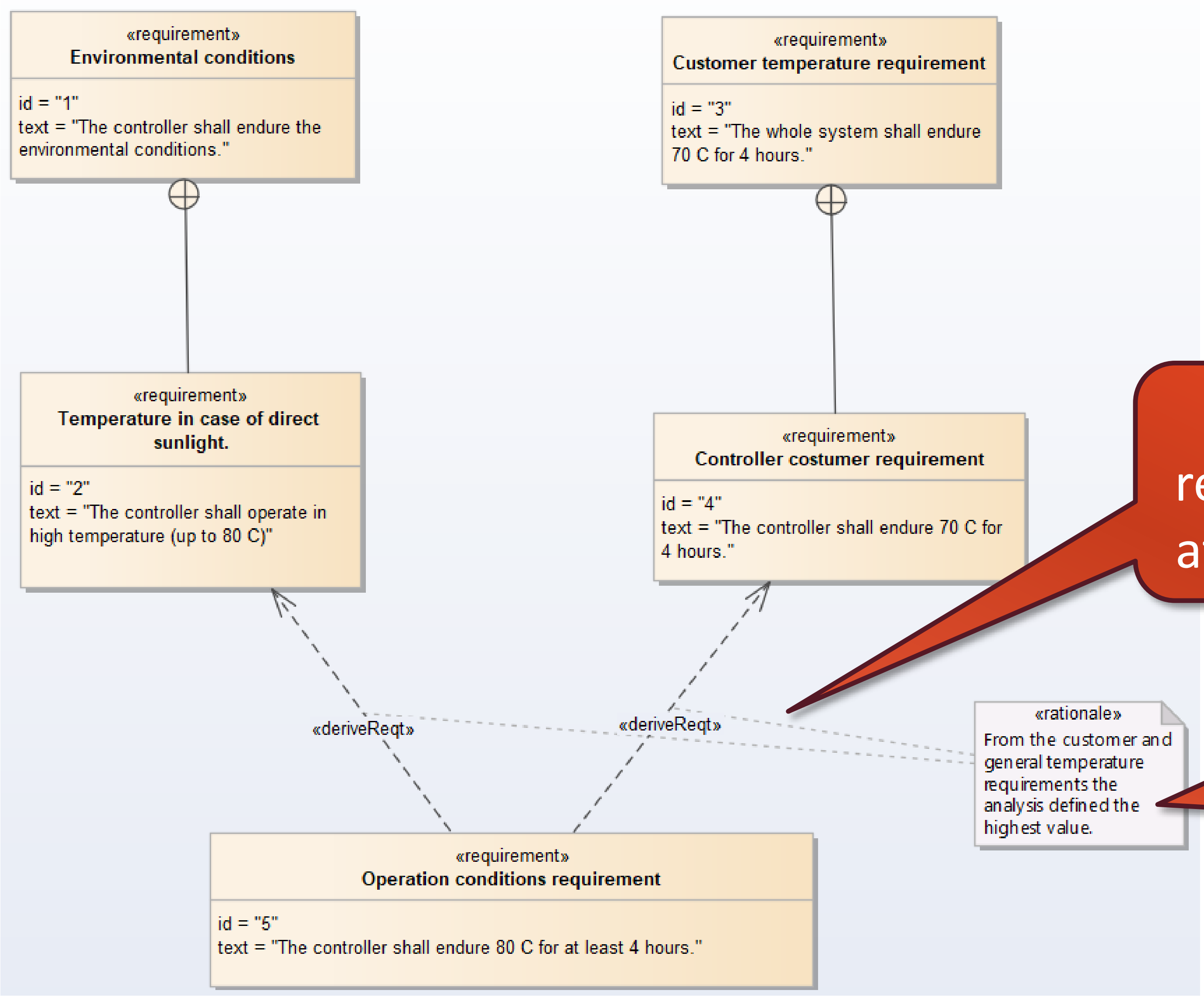


#	Id	Name	Text
1	1	Performing agricultural tasks	
2	1.1	Environment measurement	The CPAS system shall be capable of measuring the environment through its sensors.
3	1.1.1	Sensors	The CPAS uses the following sensors: temperature, humidity, luminance, rain (gauge).
4	1.2	Behavior configuration	The farmer can configure the system, when to activate its operations based on its sensor inputs.
5	1.3	Operation execution	The CPAS shall execute operations to change its surrounding environment.
6	1.3.1	Farmer notification	Whenever a notification arrives the CPAS signals the farmer based on the configured communication mean.
7	1.3.2	System shutdown	The farmer can shut down the system that immediately stops all of its active operations.
8	1.4	Supported operations	Supported operations are mowing, irrigation and spraying.
9	1.4.1	Mowing	
10	1.4.1.1	Execution of mowing	The mowing operation signals the robot lawn-mower to execute its programmed task.
11	1.4.1....	Successful mowing	If the mower robot executes its task without any problem it returns to its refueling station.
12	1.4.1....	Failed mowing	If the mower robot fails to complete its task, it sends a notification about its status.
13	1.4.1.2	Configuration of mowing	
14	1.4.2	Irrigation	
15	1.4.2.1	Execution of irrigation	The irrigation operation simply activates the pre-installed irrigation-system.
16	1.4.2....	Failed irrigation	If the irrigation-system fails, it sends a notification about its status.
17	1.4.2.2	Configuration of irrigation	
18	1.4.3	Spraying	The system shall support the control of manual spraying.
19	1.4.3.1	Initialization of spraying	The spraying operation signals the laborers to execute the spraying task.
20	1.4.3.2	Reporting of spraying	The laborers report to the system when they finished their task.
21	1.4.3....	Successful spraying	The laborers report to the system when they finished their task.
22	1.4.3....	Failed spraying	In case an error occurs during the spraying the laborers submit a form to the CPAS and it notifies the farmer.
23	2	System Maintenance	The system shall provide diagnostic information about its components for maintenance.

Requirements Trace Relations

- **Refine**
  - Depicts a model element that clarifies a requirement
  - Typically a use case or a behavior
- **Satisfy**
  - Depicts a design or implementation model element that satisfies the requirement
- **Verify**
  - Used to depict a test case that is used to verify a requirement
- **Derive**
  - Used when a requirement is derived from another requirement based on analysis
  - Typically at the next level of the system hierarchy
- **Copy**
  - Supports reuse by copying requirements to other namespaces
  - Master-slave relation between requirements
- **Trace**
  - General trace relationship
  - Between requirement and any other model element

Example derive relationship



«requirement»
Environmental conditions

id = "1"
text = "The controller shall endure the environmental conditions."

«requirement»
Customer temperature requirement

id = "3"
text = "The whole system shall endure 70 C for 4 hours."

«requirement»
Temperature in case of direct sunlight.

id = "2"
text = "The controller shall operate in high temperature (up to 80 C)"

«requirement»
Controller costumer requirement

id = "4"
text = "The controller shall endure 70 C for 4 hours."

«rationale»
From the customer and general temperature requirements the analysis defined the highest value.

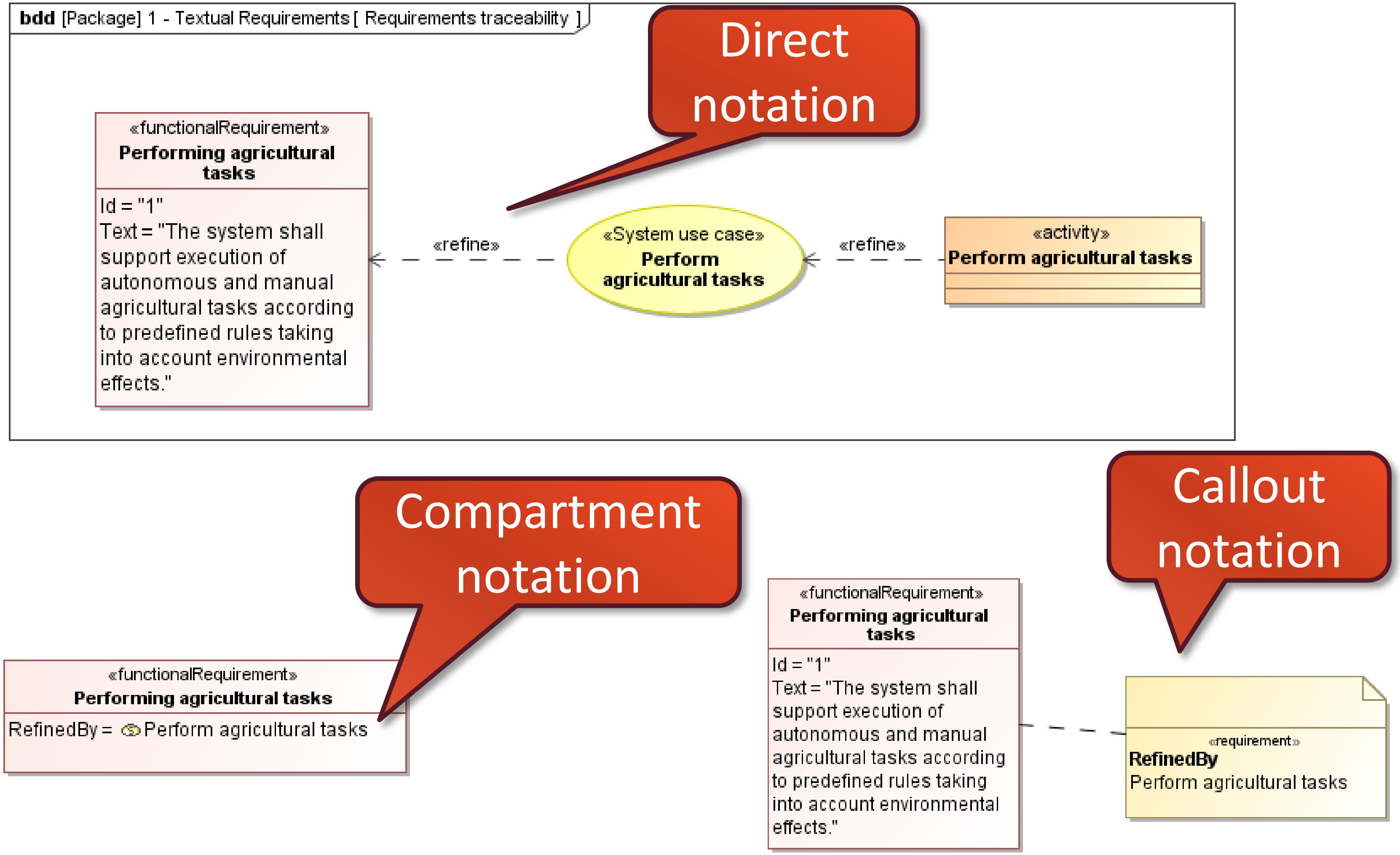
«requirement»
Operation conditions requirement

id = "5"
text = "The controller shall endure 80 C for at least 4 hours."

Derived requirement: after analysis

Rationale can be attached

Example refine relationship

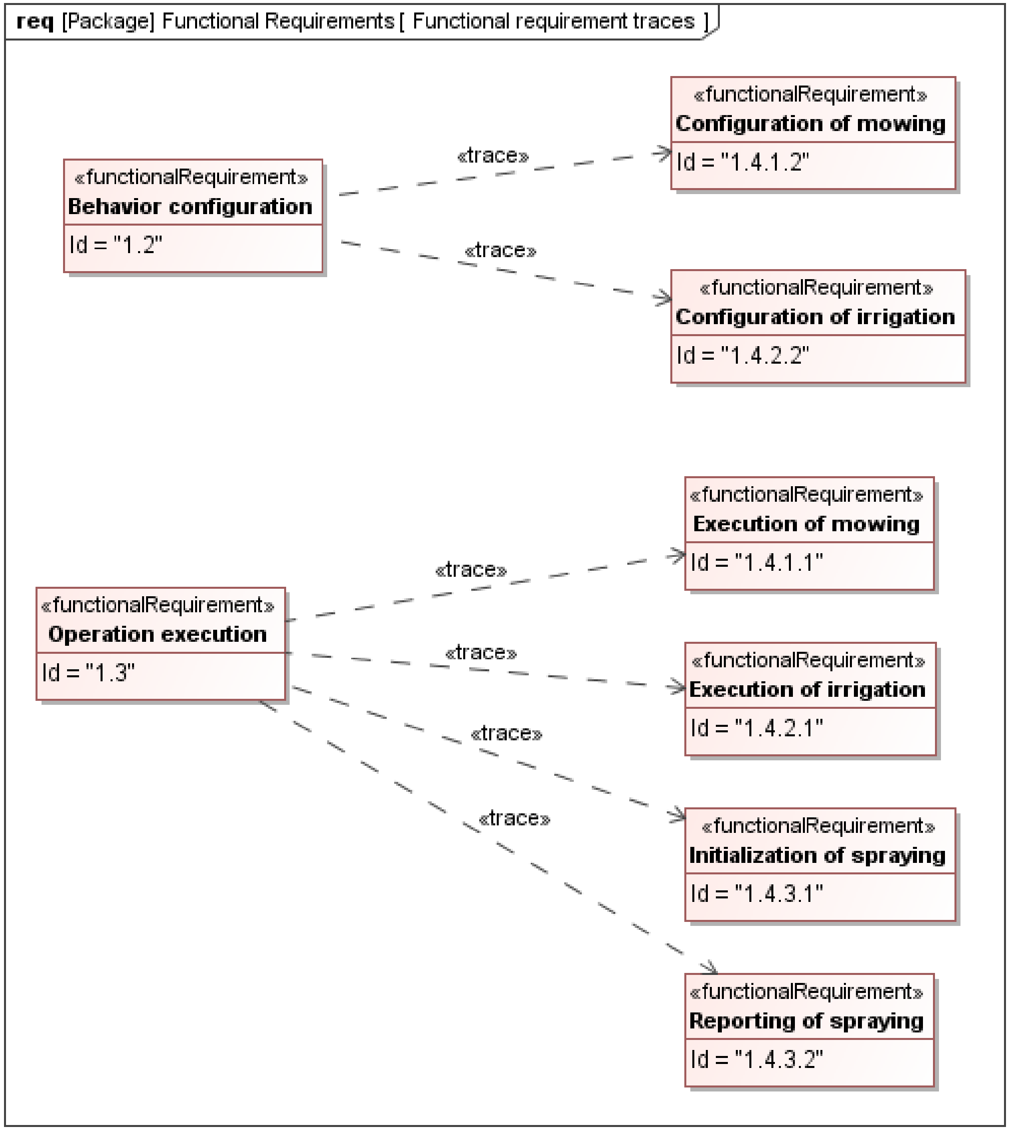

bdd [Package] 1 - Textual Requirements [Requirements traceability]
«functionalRequirement»
Performing agricultural tasks
Id = "1"
Text = "The system shall support execution of autonomous and manual agricultural tasks according to predefined rules taking into account environmental effects."
«refine»
«System use case»
Perform agricultural tasks
«refine»
«activity»
Perform agricultural tasks
«functionalRequirement»
Performing agricultural tasks
RefinedBy = Perform agricultural tasks
«functionalRequirement»
Performing agricultural tasks
Id = "1"
Text = "The system shall support execution of autonomous and manual agricultural tasks according to predefined rules taking into account environmental effects."
«requirement»
RefinedBy
Perform agricultural tasks

Direct notation

Compartment notation

Callout notation

Example trace relationships



req [Package] Functional Requirements [Functional requirement traces]

«functionalRequirement»
Behavior configuration
Id = "1.2"

«trace»

«functionalRequirement»
Configuration of mowing
Id = "1.4.1.2"

«trace»

«functionalRequirement»
Configuration of irrigation
Id = "1.4.2.2"

«functionalRequirement»
Operation execution
Id = "1.3"

«trace»

«functionalRequirement»
Execution of mowing
Id = "1.4.1.1"

«trace»

«functionalRequirement»
Execution of irrigation
Id = "1.4.2.1"

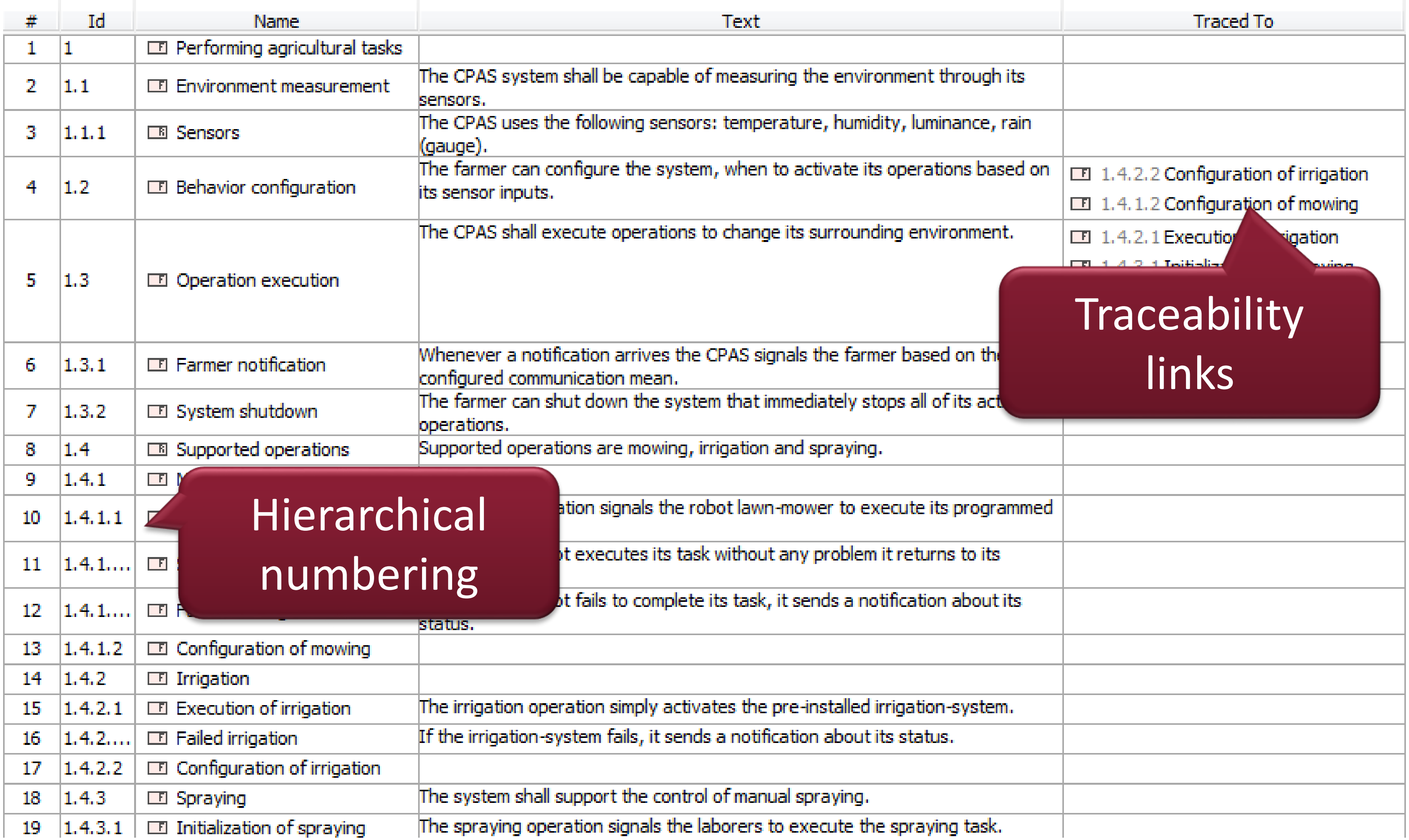
«trace»

«functionalRequirement»
Initialization of spraying
Id = "1.4.3.1"

«trace»

«functionalRequirement»
Reporting of spraying
Id = "1.4.3.2"

Requirements Relations in Table



#	Id	Name	Text	Traced To
1	1	Performing agricultural tasks		
2	1.1	Environment measurement	The CPAS system shall be capable of measuring the environment through its sensors.	
3	1.1.1	Sensors	The CPAS uses the following sensors: temperature, humidity, luminance, rain (gauge).	
4	1.2	Behavior configuration	The farmer can configure the system, when to activate its operations based on its sensor inputs.	1.4.2.2 Configuration of irrigation 1.4.1.2 Configuration of mowing
5	1.3	Operation execution	The CPAS shall execute operations to change its surrounding environment.	1.4.2.1 Executio igation
6	1.3.1	Farmer notification	Whenever a notification arrives the CPAS signals the farmer based on th configured communication mean.	
7	1.3.2	System shutdown	The farmer can shut down the system that immediately stops all of its act operations.	
8	1.4	Supported operations	Supported operations are mowing, irrigation and spraying.	
9	1.4.1	M		
10	1.4.1.1		ation signals the robot lawn-mower to execute its programmed	
11	1.4.1....	S	t executes its task without any problem it returns to its	
12	1.4.1....	F	t fails to complete its task, it sends a notification about its status.	
13	1.4.1.2	Configuration of mowing		
14	1.4.2	Irrigation		
15	1.4.2.1	Execution of irrigation	The irrigation operation simply activates the pre-installed irrigation-system.	
16	1.4.2....	Failed irrigation	If the irrigation-system fails, it sends a notification about its status.	
17	1.4.2.2	Configuration of irrigation		
18	1.4.3	Spraying	The system shall support the control of manual spraying.	
19	1.4.3.1	Initialization of spraying	The spraying operation signals the laborers to execute the spraying task.	

Traceability links

Hierarchical numbering

Summary

- Goal
  - Bridge the gap between textual requirements and requirement and design models
    - Handles textual req.s as model elements
    - Provides support for requirements traceability
- Modeling aspect
  - *What are the main requirements formulated textually and what are their hierarchy?*
- Relation of requirements to other aspects
  - Refined by model elements (e.g. use case, activity)
  - Satisfied by blocks
  - Verified by test cases