Katholieke
Universiteit
Leuven

**Department of
Computer Science**

Shared Internet Of Things Infrastructure Platform:

Domain Analysis

Software Architecture (H09B5a and H07Z9a) – Part 1

Filipcikova-Halilovic

Monika Filipcikova (r)
Armin Halilovic(r)
Academic year 2016–2017

Contents

1 Domain analysis **2**
- 1.1 Domain models . . . 2
- 1.2 Domain constraints . . . 2
- 1.3 Glossary . . . 2

2 Functional requirements **3**
- 2.1 Use case overview . . . 3
- 2.2 Detailed use cases . . . 3
  - 2.2.1 *UC1*: Name . . . 3

3 Non-functional requirements **5**
- 3.1 Availability . . . 5
  - 3.1.1 *Av1*: Database is down and a replica is used . . . 5
  - 3.1.2 *Av2*: Sensor breaks . . . 5
- 3.2 Performance . . . 6
  - 3.2.1 *P1*: Name of the quality attribute scenario . . . 6
  - 3.2.2 *P2*: Name of the quality attribute scenario . . . 6
- 3.3 Modifiability . . . 6
  - 3.3.1 *M1*: Name of the quality attribute scenario . . . 6
  - 3.3.2 *M2*: Name of the quality attribute scenario . . . 7
- 3.4 Usability . . . 7
  - 3.4.1 *U1*: Application devs upload their app . . . 7
  - 3.4.2 *U2*: Infrastructure owner changes topology . . . 8

1. Domain analysis

1.1 Domain models

This section shows the domain model(s).

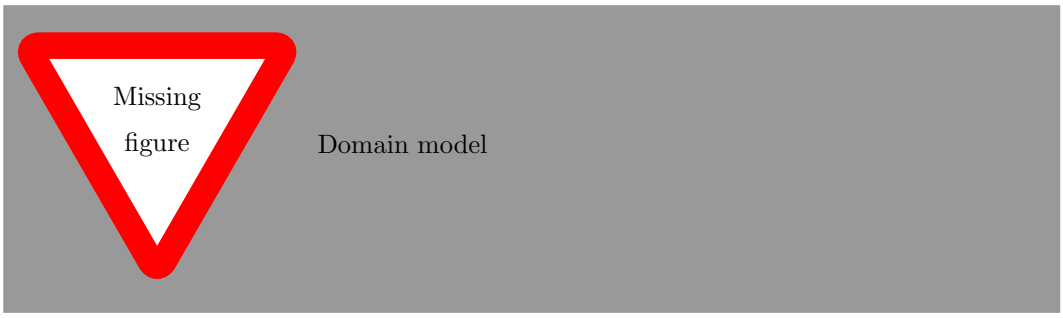

Figure 1.1: The domain model for the system.

1.2 Domain constraints

In this section we provide additional domain constraints.

- This is a first constraint.
- This is a second constraint.

1.3 Glossary

In this section, we provide a glossary of the most important terminology used in this analysis.

- **Term1**: definition
- **Term2**: definition

2. Functional requirements

Use case model

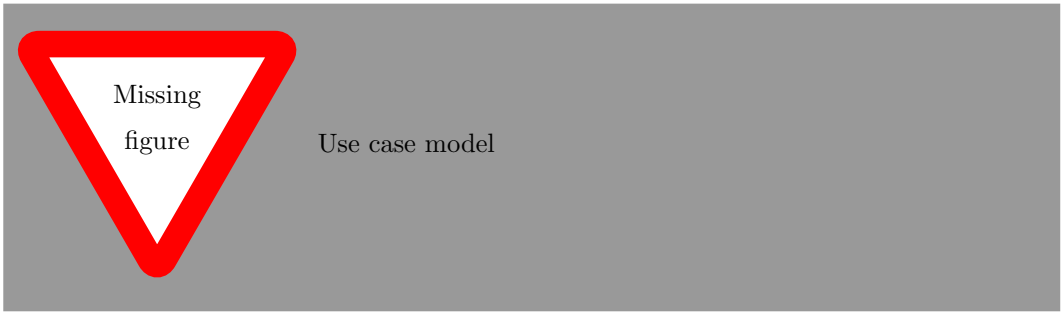

Figure 2.1: Use case diagram for the system.

2.1 Use case overview

UC1: Name Short summary of this use case scenario

2.2 Detailed use cases

2.2.1 *UC1*: Name

- **Name:** Name of use case 1
- **Primary actor:** primary actor
- **Secondary actor(s)**: secondary actor(s)
- **Interested parties:**
  - *Name of interested party:* reason why party is interested
- **Preconditions:**
  - First precondition.
  - Second precondition.
- **Postconditions:**
  - First postcondition.
  - Second postcondition.
- **Main scenario:**
  1. Step 1
  2. Step 2
  3. Step 3
  4. ...

- **Alternative scenarios:**

  3b. Alternative at step 3

- **Remarks:**
  - First remark

3. Non-functional requirements

In this section, we model the non-functional requirements for the system in the form of *quality attribute scenarios*. We provide for each type (availability, performance and modifiability) one requirement.

3.1 Availability

3.1.1 *Av1*: Database is down and a replica is used

A database in our system does not send any data.

- **Source:** External: Database server
- **Stimulus:**
  - The database crashed / does not send any response.
  - The database returns invalid data or response.
  - ...
- **Artifact:** Persistent storage
- **Environment:** Normal operation
- **Response:**
  - Use a working replica until the server can be used again.
  - If the server cannot fix the by itself, send a technician to fix the problem with the database.
- **Response measure:**
  - A working replica should be used within a range of 1s to 5s.

3.1.2 *Av2*: Sensor breaks

A sensor breaks. Another sensor is used for the responsibility of the broken one.

- **Source:** External: sensor
- **Stimulus:**
  - No data received anymore from sensor
  - Sensor is missing in heartbeat of mote
- **Artifact:** Communication channel between sensor and gateway
- **Environment:** Any state of operation, At run-time, ...
- **Response:**
  - The gateway uses another sensor to be used for the same responsibility as the broken one.
  - Report the failure to the infrastructure owner.
- **Response measure:**
  - A new sensor should be chosen within the range of 1ms to 10s

3.2 Performance

3.2.1 *P1*: Name of the quality attribute scenario

Shortly describe the context of the scenario.

- **Source:** source
- **Stimulus:**
  - Description of a first stimulus.
  - Description of a second stimulus.
- **Artifact:** the stimulated artifact
- **Environment:** the condition under which the stimulus occurs
- **Response:**
  - Describe how the system should respond to the stimulus.
- **Response measure:**
  - Describe how the satisfaction of a response is measured.

3.2.2 *P2*: Name of the quality attribute scenario

Shortly describe the context of the scenario.

- **Source:** source
- **Stimulus:**
  - Description of a first stimulus.
  - Description of a second stimulus.
- **Artifact:** the stimulated artifact
- **Environment:** the condition under which the stimulus occurs
- **Response:**
  - Describe how the system should respond to the stimulus.
- **Response measure:**
  - Describe how the satisfaction of a response is measured.

3.3 Modifiability

3.3.1 *M1*: Name of the quality attribute scenario

Shortly describe the context of the scenario.

- **Source:** source
- **Stimulus:**
  - Description of a first stimulus.
  - Description of a second stimulus.

- **Artifact:** the stimulated artifact
- **Environment:** the condition under which the stimulus occurs
- **Response:**
  - Describe how the system should respond to the stimulus.
- **Response measure:**
  - Describe how the satisfaction of a response is measured.

3.3.2 *M2*: Name of the quality attribute scenario

Shortly describe the context of the scenario.

- **Source:** source
- **Stimulus:**
  - Description of a first stimulus.
  - Description of a second stimulus.
- **Artifact:** the stimulated artifact
- **Environment:** the condition under which the stimulus occurs
- **Response:**
  - Describe how the system should respond to the stimulus.
- **Response measure:**
  - Describe how the satisfaction of a response is measured.

3.4 Usability

3.4.1 *U1*: Application devs upload their app

Application developers have built an application and wish to upload it to the Online Service. This should go smoothly.

- **Source:** Application developers
- **Stimulus:**
  - Application developers want to use the Application provider dashboard efficiently.
- **Artifact:** Application provider dashboard
- **Environment:** At normal operation
- **Response:**
  - Applications and their statuses should be displayed in a clear table.
  - The UPLOAD APP button should be pretty!
- **Response measure:**
  - The application can be uploaded with a minimal amount of clicks needed

3.4.2 *U2*: Infrastructure owner changes topology

The infrastructure owner wants to change the topology of sensors or actuators in the system.

- **Source:** Infrastructure owner
- **Stimulus:**
  - Infrastructure owner wants to use the infrastructure owner dashboard efficiently
  - Infrastructure owner wants to feel comfortable with the infrastructure owner dashboard
- **Artifact:** Infrastructure owner dashboard
- **Environment:** At normal operation
- **Response:**
  - Text is aggregated into consistent paragraphs.
  - Links and buttons have a distinct styling to make them stand out.
  - The topology is displayed clearly in a diagram.
  - There is a help system the infrastructure owner can use to learn
- **Response measure:**
  - The topology can be changed with a minimal amount of text/diagrams displayed and clicks needed to do the changes. In other words, the time do to this is bounded by X minutes.