Architectural Design Document

FT 5

October 16, 2015

Lab assignment Software Architecture

Supervisor: René Krikhaar & Wil Leeuwis

”*The architecture giveth and the implementation taketh away.*”
- Len Bass, Paul Clements, Rick Kazman

Master Software Engineering
Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Universiteit van Amsterdam

Contents

Document Status Sheet

Document status overview

General

Document title:	Architectural Design Document
Identification:	ADD_1.0.pdf
Authors:	Kevin van den Bekerom
	Kevin Bankersen
	Tom van Duist
Document status:	Final release

Document history

Version	*Date*	*Reason of change*
0.0	02-09-2015	Setup of the document layout
0.1	06-09-2015	Release version week 1
0.2	13-09-2015	Release version week 2
0.3	20-09-2015	Release version week 3
0.4	27-09-2015	Release version week 4
0.5	04-10-2015	Release version week 5
0.6	11-10-2015	Release version week 6
1.0	18-10-2015	Final Release

1. Introduction

The Architectural Design Document (ADD) will describe the architectural design steps and their corresponding rationales. It is a composition of architectural design decisions (according to the Bosch paper [1]). This document is meant to grow during the Software Architecture course and it is therefore important to check the version. In this document will create a software architecture for a robot that is meant to not only assist elderly people in their daily lives but also provide them with a form of social comfort/interaction.

1.1. Context

Care for elderly people is becoming increasingly expensive and complex. When a resident moves into an elderly home it is a known fact that many of these elderly become lonely because of a lack of social interaction with for instance their relatives or the loss of their social circle. Quality Care thinks it can improve the lives of elderly living in nursery homes. The philosophy is for every inhabitant of a nursery home to have a personal assistant making their life more comfortable and enjoyable.

We primarily focus on elderly in nursery homes compared to private homes because of the following reasons:

- Closed environment.
  - Many devices together.
  - Easier/efficient to setup and maintain.
  - Easier to conduct field tests.
  - Socially connect users who live close by trough the device.
  - Train personnel to help/educate the users to use the device.
  - Less complex/diverse environment than private homes.
  - Utilize the closed network of the nursery home.
- Reduce workload on nurses.
  - Perform simple and rudimentary tasks.
  - Allow nurse to focus on the patient.

1.2. Business case assumption

To be able to focus more clearly on the assignment, creating an architectural design, we will make the following assumption regarding the business case. This will ensure that we will not spend too much time working out the financial details and feasibility.

Nursery homes will buy the Quality Care Robot to improve the quality of life of their customers, the elderly who live there. Furthermore the Quality Care Robot will perform certain rudimentary tasks which slightly reduces the strain on the nurses which gives them more time for specialized care, which in turn improves the lives of their customers but will not reduce the amount of nurses needed. Because there is no direct monetary gain for the nursery home we will have to keep the cost low.

1.3. Stakeholders

In this chapter we describe the stakeholders involved in Quality Care. For each stakeholder we describe how they are involved in Quality Care and how this affects the architectural design.

1.3.1. Users

The group *users* are all people that actively use the Quality Care Robot. This group consists mostly of *elderly people*, since they are our target customer segment. The elderly want a robot that can help them in their everyday life, but more importantly, acts as a companion during lonely periods. Since this older generation is, in general, not accustomed to (new) technology, the Quality Care Robot should be extremely easy to operate.

Relatives of the elderly will interact with the Quality Care Robot when they visit. This interaction is expected to be minimal, since their spare visits are the reason Quality Care develops the Quality Care Robot in the first place! Therefore we will not consider this group of users anymore from this point onwards.

The medical staff has as interest being able to do their work uninterrupted. The Quality Care Robot should facilitate in that need. They might be concerned that the robot interacts with the elder in a harmful way. As a result they highly value safety.

The future brings different stakeholders, specifically users, to the forefront. In the future the Quality Care Robot might be the eyes, ears, and hands of doctors. Doctors treat elderly from a remote location, so they can treat more patients in the same time. The Quality Care Robot needs to be extremely responsive for a doctor to threat patients well.

1.3.2. Customers

The Quality Care Robot will be most effective in *nursery homes*, where there are just a few nurses treating many patients. The Quality Care Robot reduces the nurses' workload, so they can spent their time treating patients instead wasting their time on housekeeping tasks. Quality is important in health care since it reduces loneliness of patients, and enables nurses to do their work better. Concerns for nursery homes are quality, lifespan, and usability. *Insurance companies* are not a stakeholder since they are geared towards elderly people living at home.

1.3.3. Architect

The authors of this document act as architects for the company Quality Care. The architect is concerned about the complexity of the product, and whether it is future-proof. The architect's primal interest is finding the optimal balance in satisfying all stakeholders.

1.3.4. On-site maintenance

Concerns/Interests: *configurability, ease of maintenance, error handling*

On-site maintenance will work with the robot on-site. The first group of people interested in maintaining the device on-site are the *on-site mechanics*, they will perform technical maintenance on the robot and will thus be concerned with hardware configuration of the robot. The mechanic wants to be able to easily replace certain hardware components.

Another group interested in maintenance will be the nurse/pharmacist that will need to refill the robot with medicine and input the medicine-schedule. They will be mainly concerned with how easy and fast these steps will be. Configurability is another aspect the nurse wants. Each elderly person is unique, and to provide the best possible care, the Quality Care Robot should be configurable for different kind of users (tempered, passive, Parkinson patients, Alzheimer patients)

1.3.5. Quality Care - Technical

Concerns/Interests: *complexity, evolve-ability, error detection/reporting, maintainability*

The *developers* are responsible for building the system (both Java and robot programmers), this group will turn the architecture into a working system. Their main concern will thus be on the implementation of the architecture to a working system.

The *testers* are responsible for testing the system and making sure everything works as specified. Their main concern will be with the testability of the system. This could mean the implementation of a simulation program within the robot, requesting a very modular design to test individual parts etc.

The *product designer* will be concerned with how the architecture will influence their product design. They will mainly be interested in how the architecture can fit into a product that is appealing to Quality Care's customers and users.

The *maintenance* group will be responsible for bug-fixing and changing functions of the system when it has been developed (brought to the market). Their concern will thus be on how easy one can maintain the software of the robot and how updates could be provided to robots in the field.

1.3.6. Quality Care - Non-technical

Concerns/Interests: *development cost, compliance to regulation, customer experience*

The *Finance department* will be responsible for managing costs and budgets. Their biggest concern is that the project will remain within the set budget and deliver the features that have the highest return of investment.

The *Marketing/Sales department* interest is in selling the robot to the customers and to get the end users enthusiastic about using the robot (creating awareness/demand). The biggest concern for this department is that the architecture is able to deliver the features that will help sell/market the robot.

The *Risk-management department* is concerned with managing risk for the company. This department will want to know about every function within the robot to make sure the company is never at risk or that the risks are acceptable. Their main concern is compliance with rules and regulations, this could mean they want extra security, safety measures or will not allow the implementation of a certain features because their risks cannot be covered or it cannot comply with regulation.

The *training facility* is responsible for teaching the users (nurses/elderly) how they use the robot within a nursery home. Therefore they will mainly be concerned with how easy it will be to train users in using the robot through for instance a simulated environment.

1.3.7. Manufacturing

Concerns/Interests: *hardware complexity, hardware availability, assembly time*

Complexity of manufacturing the robot should be manageable, this directly translates into the availability of the different hardware components and the complexity and time needed for the assembly process all of which directly influence the manufacturing cost which should of course be as low as possible.

1.3.8. Future stakeholders

Concerns/Interests: *accessibility*

third party application developers can make sure the Quality Care Robot stays up to date with the latest features. The Quality Care Robot should then contain a software framework similar to the Apple app store, geared towards robot functionality.

2. System overview

2.1. High level requirements

This chapter will describe the high level requirements of the Quality Care Robot. This will outline, on a highly abstract level, what the system should ultimately be capable of. Further in this document the focus will be on the first three high level requirements.

2.1.1. Medical assistance

Medical assistance is the most important aspect of the Quality Care Robot. People in the target segment on average have to take a couple of medicine pills per week, but multiple pills on different occasions of the day is not uncommon. To assist with this the system could remind the user, or even dispense the pills to the user, on the scheduled times. The system could also monitor the insulin levels of sugar patients and analyze blood pressure. By using a wearable extension it could also be possible to monitor an inhabitants vital functions like heart rate, temperature and oxygen levels.

2.1.2. Social interaction

The *social interaction* is the second most important aspect of the Quality Care Robot. The user should be able to communicate with the system trough voice commands to strike up simple conversations and ask for certain information, such as the weather or news stories. Getting into contact with friends and relatives is also an important part of the system, through voice calls and the scheduling of appointments it can be a gateway into a more active social life. The system should also be able to recognize the voice of the main user and react to certain emotions. For instance if the user is yawning the system could respond to that, or if the user looks sad and has had poor social interaction for a while it could schedule an appointment with a relative or call a friend.

2.1.3. Support in everyday life

The third aspect of the Quality Care Robot is *support in everyday life.* Many simple and tedious tasks should be automated through the system or be carried out trough voice commands. The system could be used as an agenda/scheduler, alarm clock, control lighting, doorbell/intercom etc. The thermostat could also be controlled by the system through a simple schedule in conjunction with the schedule of the user. Other more advanced features could be vacuuming and assisting with simple kitchen tasks such as making coffee.

2.2. Main functionality

In this section we describe the main functionality of the Quality Care Robot. The goal of this section is the give the reader a sense of what the Quality Care Robot can do when it is fully build. In Appendix E the full mind map is given, which includes all functionalities we taught of. Some functionalities in the mind map that are not listed here are considered to be good features for future versions of the Quality Care Robot. We will therefore take into account these functionalities as well in constructing the ADD. For prioritizing, the Moscov model is used [12]:

M : Must be implemented. Without this the product has failed.

S : Must be implemented. If not, stakeholders are dissatisfied.

C : Not essential for the product to be a success.

W : Wont have. Nice features for future versions.

2.2.1. M - Provide the right pills at the right time

The Quality Care Robot is able to provide pills to the user as prescribed by the doctor.

2.2.2. M - Contact relatives/nurses

The Quality Care Robot can contact relatives on demand, to make a visiting appointment for example. The robot can contact the nurse as well when the user is in immediate need of assistance.

2.2.3. M - Detect emotional state

The Quality Care Robot is able to correctly detect and store *basic* emotions, relating them to a limited set of events. Basic emotions include anger, sadness, happiness and neutral state. Speech recognition and facial recognition are used in conjunction to increase the reliability of emotion detection [5]. This also entails recognizing speech and faces.

2.2.4. M - Health monitoring

The Quality Care Robot is able to track a user's heart rate and blood pressure.

2.2.5. S - Hold simple conversation

The Quality Care Robot is able to maintain simple conversations with the user. Covered topics are the weather and the status of different users within the Quality Care Robot network of the nursery home. The Quality Care Robot is able to suggest meetings with different Quality Care Robot users based on previous experience. A concrete example is suggesting a game of cards with Mrs. Jansen After the game the Quality Care Robot asks how the game went. Mrs. Roberts reacts very angry. The Quality Care Robot registers this emotion and suggests a different card partner the next time.

2.2.6. S - Video monitoring

The Quality Care Robot will show (secure line) video in the QC control center located in a room in the nursery home.

2.2.7. C - Home automation

The Quality Care Robot has a framework that enables enabling or disabling different kind of home automation modules. For now these include a module to answer and open the door, and a module to serve coffee or tea.

2.3. Cloud Robotics

We clearly divided the system in two subsystems. One for the Quality Care Robot itself, and one being the central processing unit, that forms the *brain* of all robots. This setup is called cloud robotics 2.1. IEEE on Cloud Robotics [9], the NAO Robot[10] and Earth Robot [11]

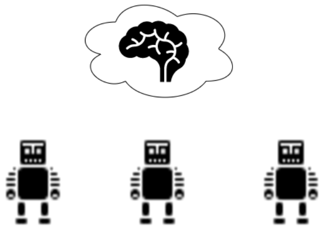


Figure 2.1.: Cloud Robotics

2.3.1. Environment

A nursery home is a perfect place for Cloud Robotics:

- Controlled environment
- Possible to provide stable (WiFi) network (air mesh networking).
- Easy to implement and maintain a central computer.
- More cost efficient.
- Easier security, the robots do not need Internet, only the server does. Meaning it will be easier to secure one server then a multitude of robots.

2.3.2. Robotic advantages

Robots are very complex machines that require specialized hardware and software to control it. Cloud robotics would allow for:

- Simpler and cheaper robots.
- More adaptable robots.
- Smarter Robots.
- Easier to maintain
- Increased battery-life due to needing less processing power.
- Faulty robots can be replaced and will require less configuration.

2.3.3. Single control unit

- Hardware maintenance is cheaper, faster and easier to understand (well understood).
- Extra processing power can be added locally or an external "Cloud" could be used
- Smarter Robots
- Easier to maintain/expand (software updates)
- Increased battery-life due to needing less processing power.
- Easier to keep data and systems redundant.
- More dynamic (scaling)

2.3.4. Risks

Using a central brain does pose some risk, mainly network related.

- Risk of losing connection , what does the robot do then (Will it stop/can it do rudimentary things)
- Network latency
- Security, if the server gets hacked one would have access to all the robots.

2.3.5. Example scenarios

One of the big current technology trends is using big data analysis and machine learning. It is especially believed that within the medical field big data can cause a revolution in treating, discovering and preventing diseases. A downside of big data and machine learning is that it requires great computing power, much more than our robot could possibly provide. But by using cloud robotics we can use our robot as a data-collector and let the server side handle the calculations, the only thing our robot is required to do is to gather as much information and data about the user as possible. An example is a persons heart rate, by collecting as much heart rate data as possible and compare it with other people it will become possible to predict a persons health state, mixing this with things like voice vibrations the system could possibly say something about a persons vitality and or mental state and will thus allow caretakers to improve a persons quality of life. A higher quality of life is associated with lower medical cost and less diseases etc (CDC).

Another example of cloud robotics is using the server side to manage home automation, for instance opening doors. By using a server for this task it is possible to encapsulate the doors from the robot. Instead of telling every single robot which door it can control one simply needs to configure the server. Since the robot uses voice commands to control a persons environment the robot only needs to record what the user said and send this to the server, voice analytics on the server will then be able to recognize which door the user wants to open and calls the function internally. This approach will not only make the robot simpler, it will also be much easier to test (instead of every robot only the interface of the server needs to be tested).

3. System decomposition

This chapter describes the decomposition of the Quality Care Robot system in different modules or parts. Architectural views are used to decompose the system according to different (usage) scenario's. Architectural patterns and styles describe a set of tactics to satisfy quality attributes of several large subsystems and can be found in Appendix C. The Monitoring Module is worked out in detail regarding needs, quality attributes, metrics, quality scenario's containing a certain measurable that is met by using a set of tactics.

3.1. Architectural views

The architectural views will be distilled by using the 4+1 view [13].

3.1.1. Logical view

The logical view (figure: 3.1) describes the behavior and functionality of the system in relation to the user. As this view focuses on the end user many parts of the system have been simplified to focus on the parts that are important to the end user(s).

The main user of the Quality Care Robot, the elderly, interacts with the system primarily by means of speech, but the system also uses image as input. This input will be digested by the IO (input output) modules (not depicted in the figure) located on the robot and uses the central server in the Nursery Home to digest the input. When the input has been digested the commands will be processed by the Quality Care Robot, this will either trigger an event, e.g. medication or a beverage will be dispensed, or there will be a response through the IO modules who will notify the user either by means of sound or trough the display.

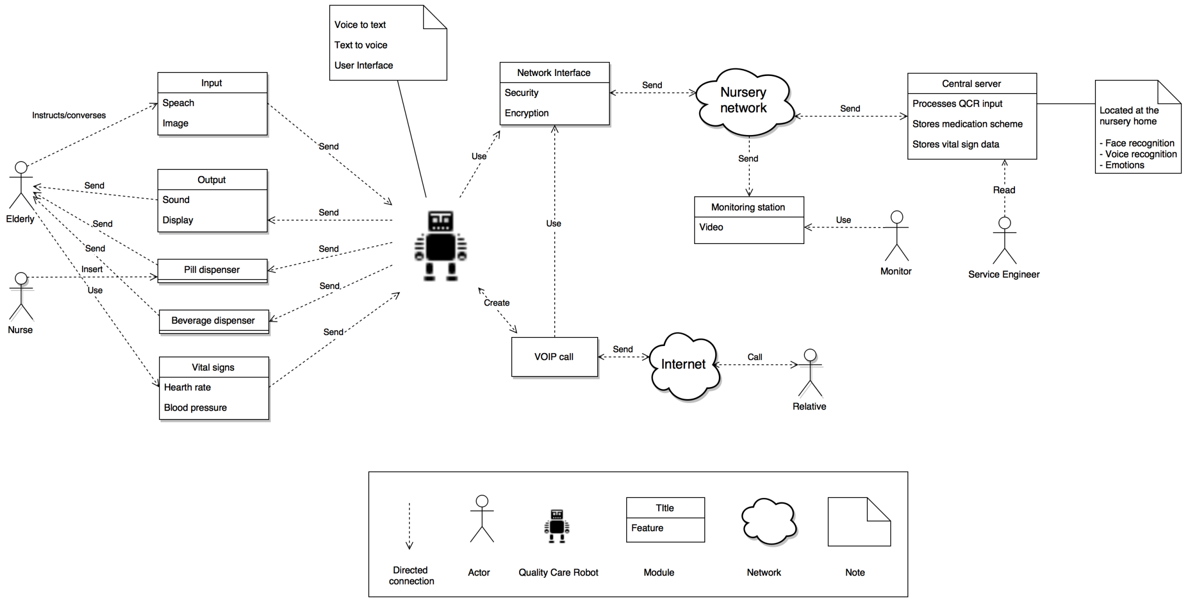


Figure 3.1.: Logical view for robot and user functionality.

3.1.2. Process view

The process view describes the runtime behavior of the system. The major systems are depicted in Figure 3.2. Two main processes can be distinguished; the *robot controller*, and *server controller*. The robot controller consists of several subprocesses (*Medical Management, Monitoring, I/O*) that together constitute the functionality of the robot. *Movement* (rotating on a stationary platform) is a separate, computationally intensive, process that communicates with the robot controller, but has its own resources. The reason being not to endanger medically critical processes (*Monitoring*, *Medical Management*).

The server controller can spawn 3 types of processes. *Conversation* for generating conversation lines. *Emotion* for calculating an user's emotion. *Sceduler* to make appointments between users. The server controller can spawn multiple Emotion processes for instance, based on demand. It uses third party services for facial and speech recognition.

The robot controller can request services from the server controller, or send video or medical information which can be processed by the server controller.

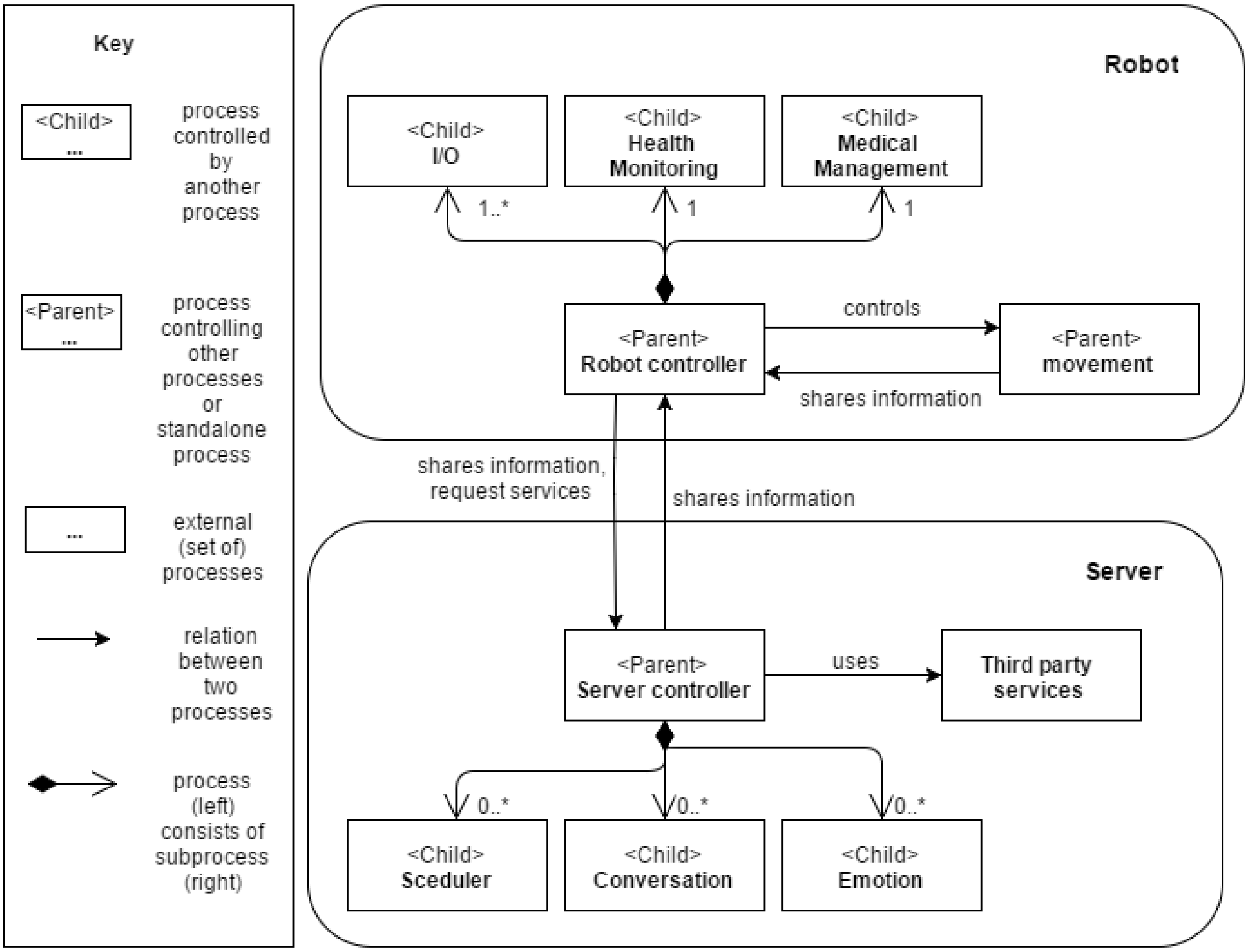

Key
<Child> ...
process controlled by another process
<Parent> ...
process controlling other processes or standalone process
...
external (set of) processes
relation between two processes
process (left) consists of subprocess (right)
Robot
<Child> I/O
<Child> Health Monitoring
<Child> Medical Management
1..*
1
1
<Parent> Robot controller
controls
<Parent> movement
shares information
shares information, request services
shares information
Server
<Parent> Server controller
uses
Third party services
0..*
0..*
0..*
<Child> Sceduler
<Child> Conversation
<Child> Emotion

Figure 3.2.: Process view of complete system.

3.1.3. Implementation view

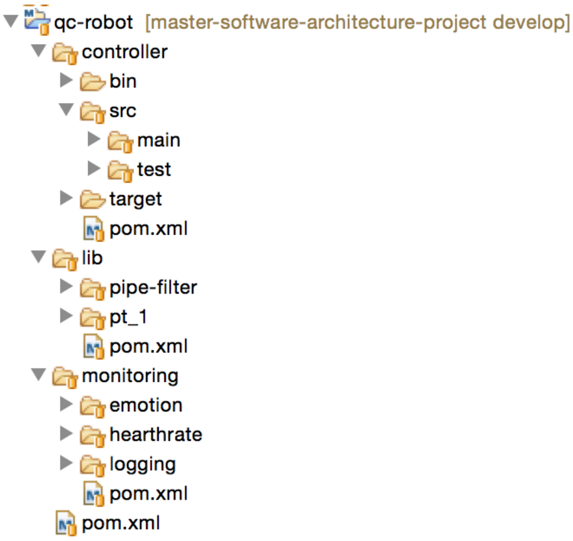
Java will be used as the coding language for this project, Maven will be used as build tool and used for project management, this gives a default project structure. This structure is chosen to simplify the way builds can be run.

Programming

- Programming language is *Java.*
- Present *coding standard* must be followed.
- *Maven* must be used as a build tool.
- Only use the libraries provided or make a request for a new library to be added.

File locations

All the components of the project should be contained in the QualityCareRobot folder. This folder will contain all sub-components in their designated folder, see fig. 3.3 for an example. Every module of the system. e.g. monitoring, will reside in the root folder and contains the components that makes up this module. Reusable components that are used by multiple other components, e.g. third party libraries or in-house developed patterns, will reside in the *lib* folder. This will ensure every component resides in a logical place within the repository, it will also be able to extract or replace loosely coupled components.



```
▼ qc-robot [master-software-architecture-project develop]
  ▼ controller
    ▶ bin
    ▼ src
      ▶ main
      ▶ test
    ▶ target
      pom.xml
  ▼ lib
    ▶ pipe-filter
    ▶ pt_1
      pom.xml
  ▼ monitoring
    ▶ emotion
    ▶ hearthrate
    ▶ logging
      pom.xml
    pom.xml
```

Figure 3.3.: Project structure example.

- **Component**

  /doc Contains all the documentation

  /src/main/Java Contains all the actual Java code.

  /src/main/resources Contains all the resources used by this component.

/src/test/Java Contains only the test files (J Unit).

/src/test/resources Contains the resources used by the unit tests.

- **lib**, either:

  Component Our own implementations or third party sources use the component structure.

  Component Third party jar:

  /doc Contains the provided documentation.

  component name.jar The compiled .jar file containing the source code.

Version Control System

For version control GIT will be used. A .git ignore file is present; this file will exclude every file in the repository that matches a specified file extension or naming pattern.

Release and version management

For version management we will use GIT flow [6]. This is a development model which describes a version management scheme. Many different git clients such as Source Tree have build in support for working with GIT flow. 3.4 provides a schematic overview of how the GIT flow branching model should be used.

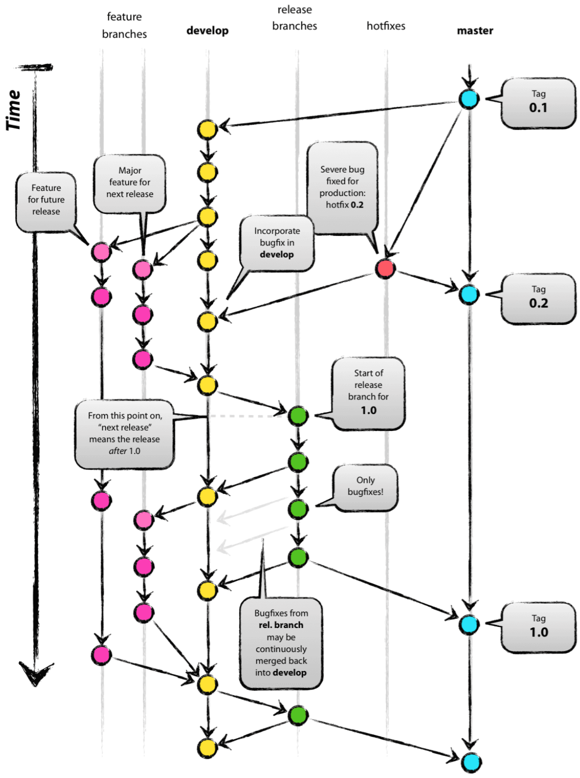

feature branches
develop
release branches
hotfixes
master
Time
Tag 0.1
Feature for future release
Major feature for next release
Severe bug fixed for production: hotfix 0.2
Incorporate bugfix in develop
Tag 0.2
Start of release branch for 1.0
From this point on, "next release" means the release after 1.0
Only bugfixes!
Bugfixes from rel. branch may be continuously merged back into develop
Tag 1.0

Figure 3.4.: Git branching

Development Environment

For development Eclipse will be used in combination with Maven. This combination has been chosen because of the great integration of Maven into Eclipse, especially dependency management.

Libraries

Developers can make a request for a specific library, this library could then be added to the build tools. A couple of standard libraries will be provided. This includes but is not limited to J Unit 4.*, Log4j 4.*. The architect has the final say which libraries can be used to ensure consistency and compatibility between developer teams.

3.1.4. Deployment view

The deployment view maps runtime processes to their hardware components. It should give an overview of the different hardware components that make up the system and which process run on which hardware component.

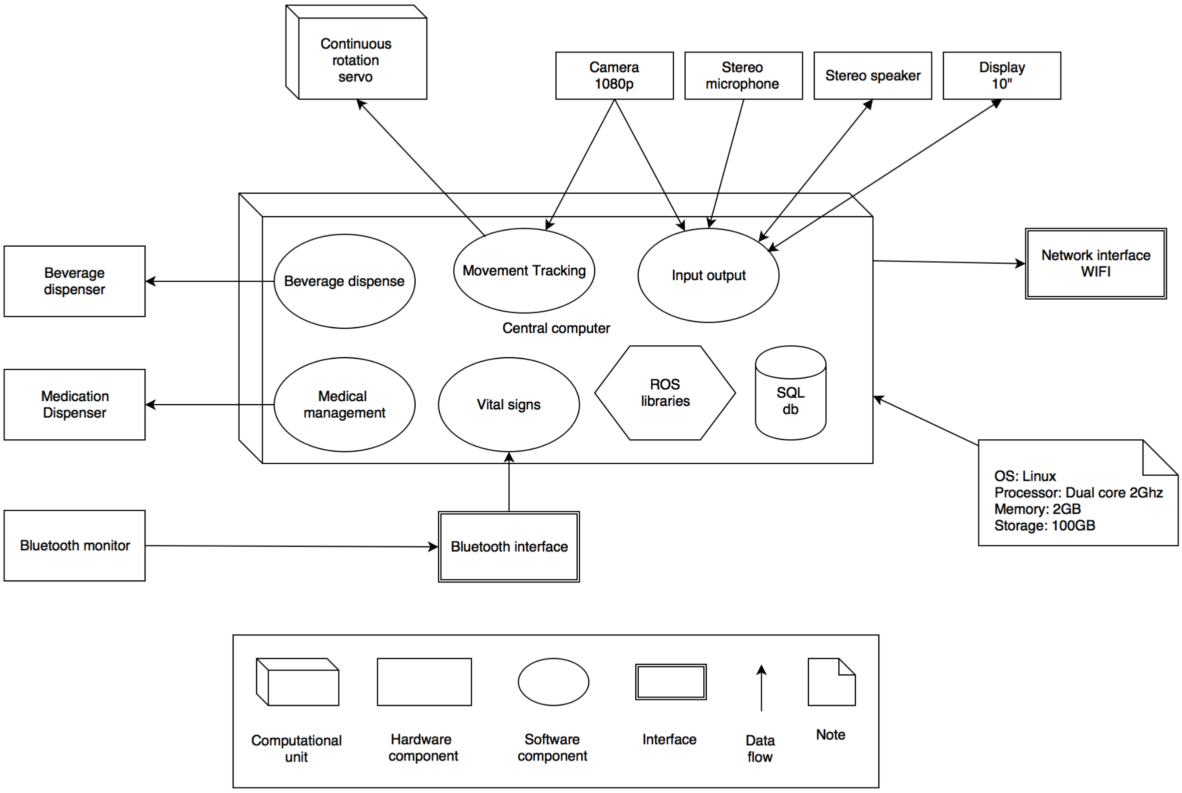


Figure 3.5.: Deployment view for Robot processes and hardware.

The robot comprises of a main computer which runs Linux, on which all of the core processes run. Reason for using Linux is that it can run an OS specifically designed for robots (ROS) [7]. ROS will allow us to easily implement features like camera tracking since this is part of their library. Using ROS will therefor make development easier and since this package is frequently used in the field it will not be hard to receive support.

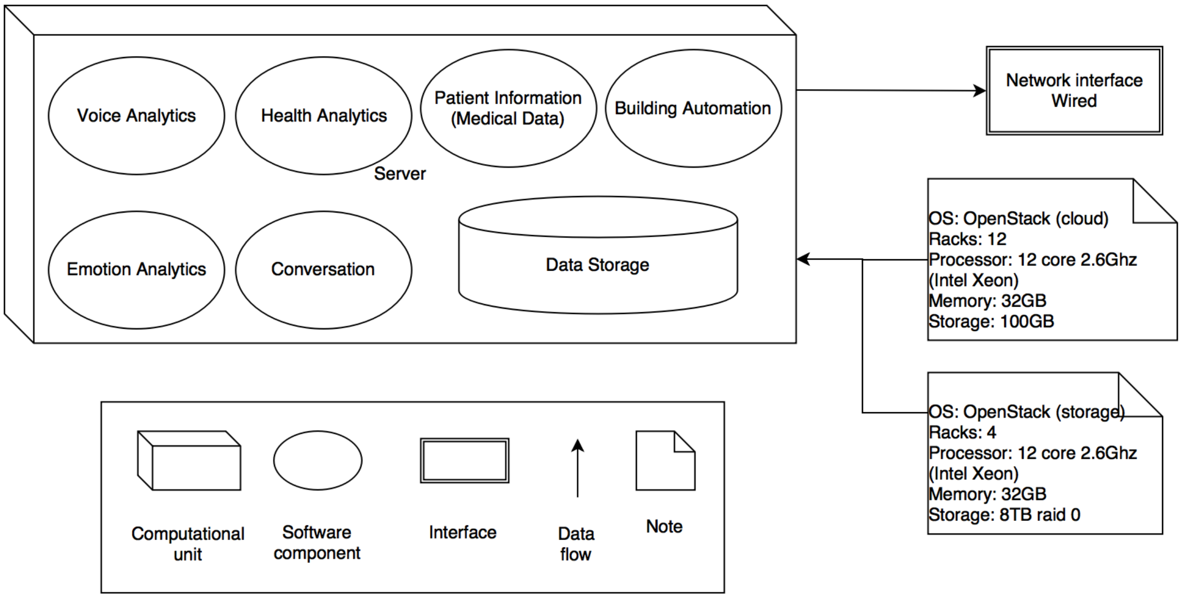


Figure 3.6.: Deployment view for the server.

The server comprises of multiple racks that have different responsibilities. Part of the racks is used for data-storage, the others are used for computational tasks. We choose to use Openstack [8] for this since it will allow us to create a private cloud environment (linking all the physical units into a big virtual machine). The processes running on the server are mainly building blocks that are bought from an external vendor.

3.1.5. Use Case view

The Use Case view describes key architectural scenarios between the interaction of the different users and the system. We will first give a diagram with the different use cases and then explain the different cases in detail.

Use Case diagram

Below you find a use case diagram that gives a brief overview of the different use cases that comprises the system. It explains which actors interact with the system in the different scenarios.

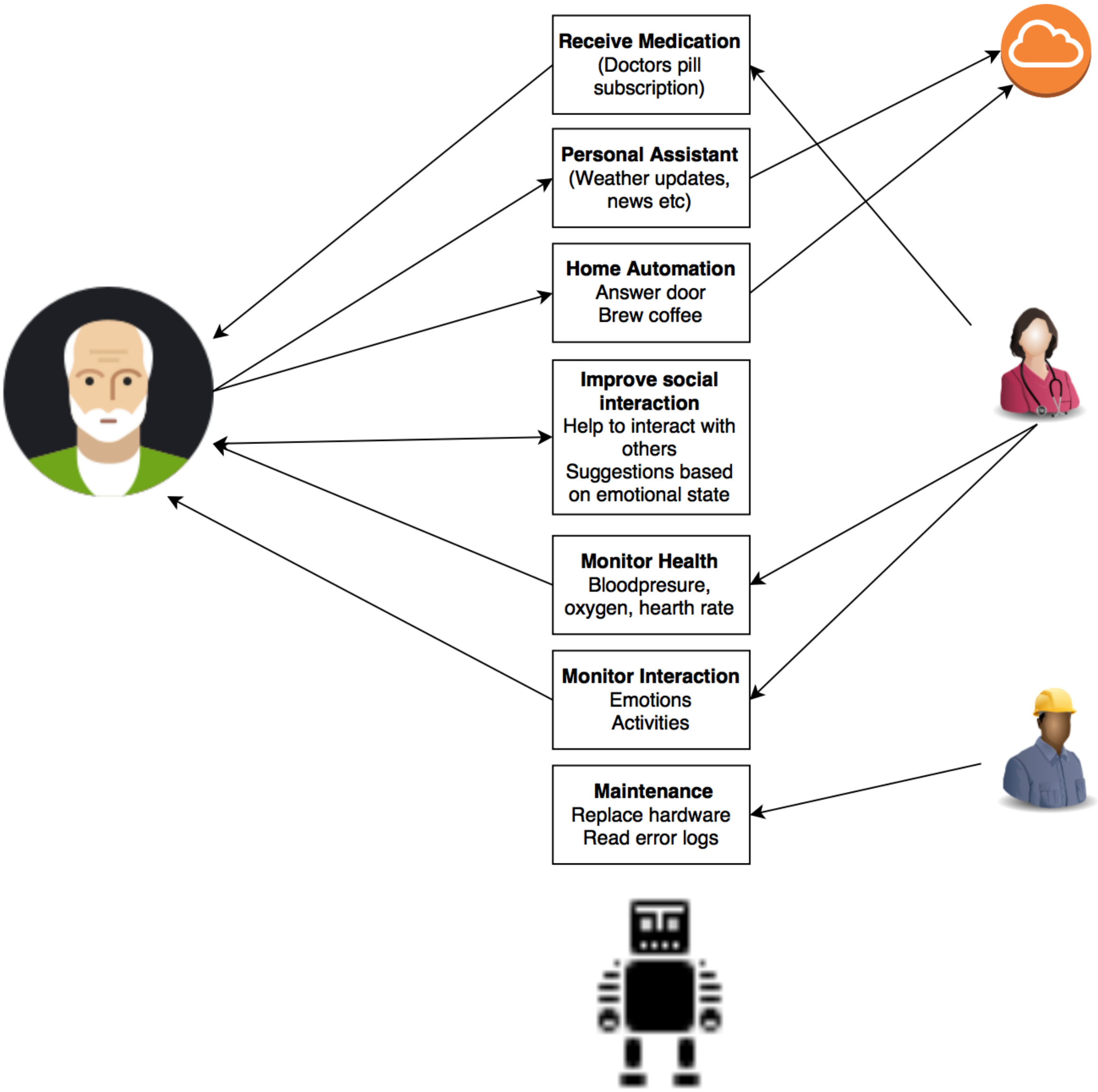

Receive Medication
(Doctors pill subscription)
Personal Assistant
(Weather updates, news etc)
Home Automation
Answer door
Brew coffee
Improve social interaction
Help to interact with others
Suggestions based on emotional state
Monitor Health
Bloodpresure, oxygen, hearth rate
Monitor Interaction
Emotions
Activities
Maintenance
Replace hardware
Read error logs

Figure 3.7.: Use case diagram giving an overview of the different use cases that comprises the system.

Use case: medication management

This section describes the different use cases with regard to the medication management system.

Name: *The elder takes his medication.*

Initial state The Quality Care Robot has a recently updated medication schedule and it is time for the next dose of medication.

1. The Quality Care Robot notifies the elder that it is time for his medication by means of voice.

2. The elder approaches the Quality Care Robot, looks into the camera and says a few words.

3. The Quality Care Robot authorizes the elder.

4. The Quality Care Robot notifies the user about the successful authentication and that it will dispense the medication.

5. The Quality Care Robot dispenses the medication.

6. The elder takes his medication.

3a The Quality Care Robot fails to authorize the elder.

- **.1** The Quality Care Robot notifies the user about the failed authorization and instructs the user to state his name and look into the camera.
- **.2** The elder may either comply (return to step 3) or abort by expressing that he does not wish to have his medication (go to step 3b.2).

3b The Quality Care Robot recognizes the user as a different person than the one the medication is meant for.

- **.1** The Quality Care Robot notifies the user about the mismatch in the authorization.
- **.2** The system notifies the nurse about the failure to dispense the medication.

Name: *The nurse fills the medication dispenser.*

Initial state The pill dispenser is empty once a week as it contains medication for one week only.

1. The nurse approaches the Quality Care Robot and states that she wishes to replace the pill dispenser cartridge.

2. The Quality Care Robot authorizes the nurse as someone with proper permissions.

3. The Quality Care Robot unlocks the current pill dispenser cartridge.

4. The nurse removes the old cartridge and inserts the new one.

5. The Quality Care Robot detects that the new cartridge is inserted and notifies the nurse.

2a The Quality Care Robot fails to authorize the nurse or the validated user has not the proper permission.

.1 The Quality Care Robot states that he was not able to properly identify the nurse (go back to step 1).

Use case: suggest social activity

Name: *Suggest social activity.*

initial state Upon morning analysis the system notices that the user has not left the room for a prolonged amount of time in the last few days.

1. The Quality Care Robot asks the elder if he has any special plans this afternoon.

2. The elder responses negatively.

3. The system asks the elder if he might like to play a game of cards with the neighbor.

4. The elder responses that he would like that.

5. The Quality Care Robot establishes a call with the neighbor.

6. The elder and the neighbor discuss a time at which to play cards.

2a The elder expresses that he things on his mind today.

.1 The system responds by wishing the elder a good day.

4a The elder responses by saying he does not fancy a card game.

.1 The Quality Care Robot suggests a different game (return to step 4).

3.2. Monitoring Module

The heart rate monitoring system comprises of multiple different system, we will describe the needs of the most important stakeholders with regard to these systems. We then work out the details of two of such systems according to these needs and subsequent measurable goals.

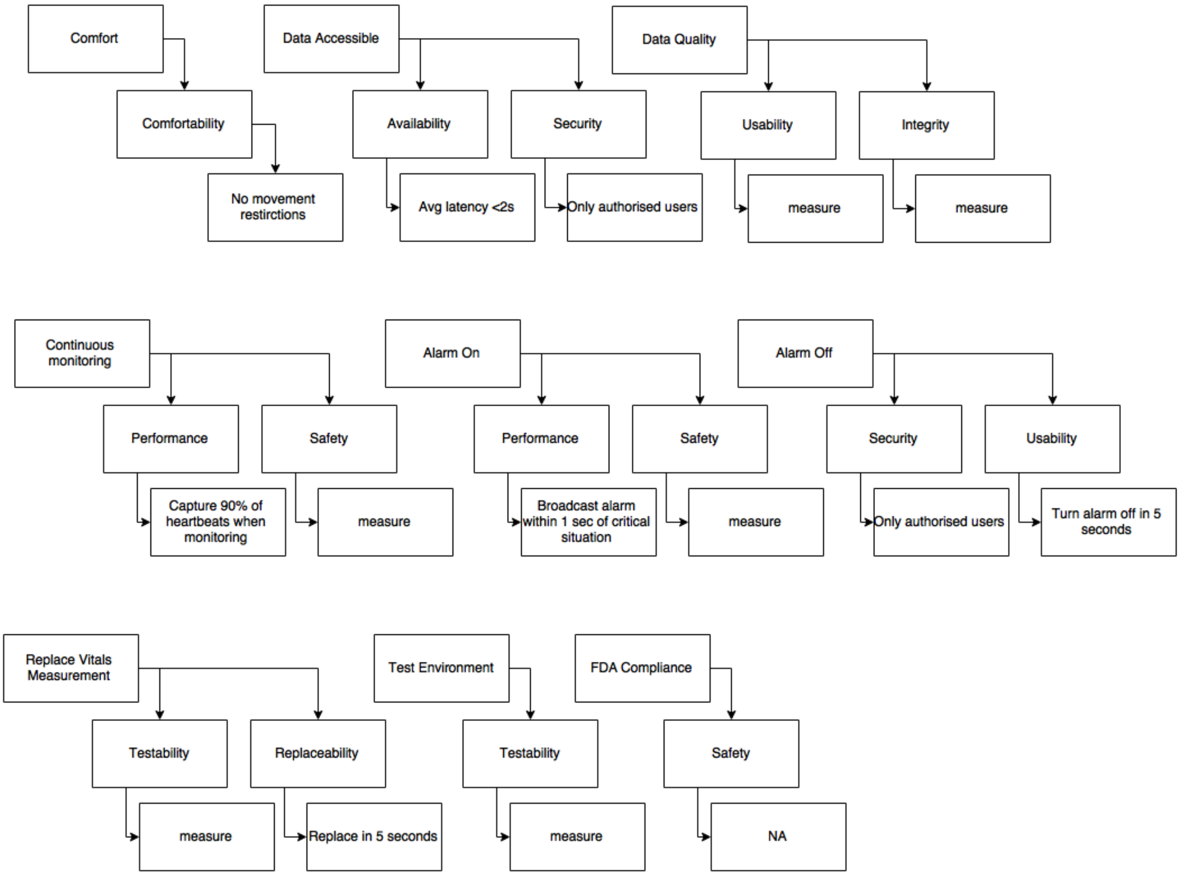


Figure 3.8.: Needs and qualities health monitor system.

3.2.1. Needs

Key: *Need (Stakeholders having said Need)*

- Comfort (Elderly)
  - The user wants a comfortable experience using the system.
- Data accessible (Nurse, Doctor)
  - The data must be accessible by the nurse or doctor.
- Data quality = right data accessible (Nurse, Doctor)
  - When the nurse or doctor accesses the monitored data they need to be reasonably sure that the data can be trusted.
- Continuous monitoring (Nurse, Nursery Home)
  - The nurse and nursery home want to be notified fairly quick when the health of a patient degrades.
- Durable components (heart rate monitor, blood pressure meter) (Nursery Home)
  - The nursery home does not want to replace the hardware often.
- Sound alarm when patient has a cardiac arrest (Nursery Home, Nurse, Doctor)
  - The people responsible for the health of a patient need to be notified of a cardiac arrest.
- Vitals measurement devices easily replaceable (Nurse)
  - When a hardware monitor is broken, the nurse needs to replace one fairly easily.
- Modular, open interface (Developers)
  - The developers want a hardware system they can easily develop upon.
- Different test environments, single module testable (Testers)
  - The testers need to test the system under different conditions.
- Easily configurable (Training facility, On-site maintenance, Nurse)
  - The system must be easily configured for each user by the nurse.
- FDA compliance (Nursery Home, Risk Management)
  - The system needs to adhere to the safety regulations of the FDA.

3.2.2. Quality Attributes

Comfort (**Comfortability**)

- Is the monitoring device comfortable for the patient?

Data accessible (**Availability, Security**)

- Patient data should be available when needed.
- Patient data should be secure, only a certain group should have access.

Data Quality **(Usability, Integrity**)

- The produced data should be usable.
- The produced data should be accurate.

Continuous monitoring (**Performance, Safety**)

- The output of a monitoring device should be handled asap.
- The system should have the patients health as it top priority

Durable components - This is a vague hardware requirement so will be left out.

Sound alarm (**Performance, Safety**)

- When assistance is needed this should arrive within a certain time.
- Patients safety should be the top priority.

Turn alarm off (**Security, Usability**)

- Only nurses or doctors should be able to disable the alarm.
- An alarm should be easy to turn off.

Replace vitals measurement device (**Testability, Configurability**)

- When a device must be replaced testing the device is important
- A new device should only require minimal configurablility.

API (**Development Distributivity**) - By default enforced by the hardware separation.

Test environment (**Testability**)

- Testing should be easy, both during development and during prototyping.

Configuration (**Configurability**)

- Installing a device might need configuration.

FDA compliance (**Safety**)

- There is no regulation for non life-critical monitoring systems, the closest thing are mobile monitoring systems. These are not regulated [14].

3.2.3. Measurable's

The following measurable's are derived from the quality attributes.

Performance

- Capture 90% of heartbeats per minute. When the system is monitoring, it should be able to capture and process enough data to be accurate. When capturing for instance 54 seconds out of a minute you have a very accurate average for the BPM, this reduces the chance for error to nearly non-existent [18].
- Nurse must receive the alarm of a cardiac arrest within 1 minute of the situation being discovered. The first 6 minutes in a cardiac arrest are critical [19], it is the responsibility of the nurse to get to the patient within five minutes, we believe this to be sufficient since nursery-homes tend to have a nurse stationed at every floor.
- Within one second of detecting a cardiac arrest an alarm message must be broadcasted. When a cardiac arrest is detected this should nearly instantaneously be delegated to the systems responsible for handling this alarm.

Availability

- Measure heart rate on average at least for 23 hours every day. To increase the chance to detect a cardiac arrest the monitoring time has to be maximized, but it is also realized that monitoring 100% of the time is unfeasible for example when the user takes a shower. An other reason for monitoring the elderly for as long as possible is that analyzing their heart-rates and in the future possible other vital signs can provide intelligence in developing sicknesses, mental health and the vitality of elderly through (big) data anlysis.
- Personnel must be alarmed of a cardiac arrest 99% of the time that such a situation arises (i.e. 1% of cardiac arrests may slip through the system). Ideally this should of course be 100% but that is not realistic, also compared to the current situation (no monitoring) this number is a huge increase.
- When processing and storing the historical heart rate data, 5% may be lost due to server overload or data corruption. As it is not critical to save all of this data to get a picture of the history of a patient's health.

Replace-ability

- A measurement device must be replaceable by a nurse in less than 5 minutes. This increases the monitoring time and ensures the nurse does not lose valuable time.

Usability

- Turn of the alarm within 5 seconds. When the alarm of a critical health situation (cardiac arrest), authorized personnel should be able to turn off the alarm with minimal effort.

Safety

- Only authorized users can turn off the alarm. When the alarm of a critical health situation (cardiac arrest) is sounding it can only be turned off by authorized personnel (i.e. Nurse, Doctor).

3.3. System 1: Heart rate collection

In depth description of the heart rate collection system. For an overview of the system see figure: 3.21

600 **Scenarios**

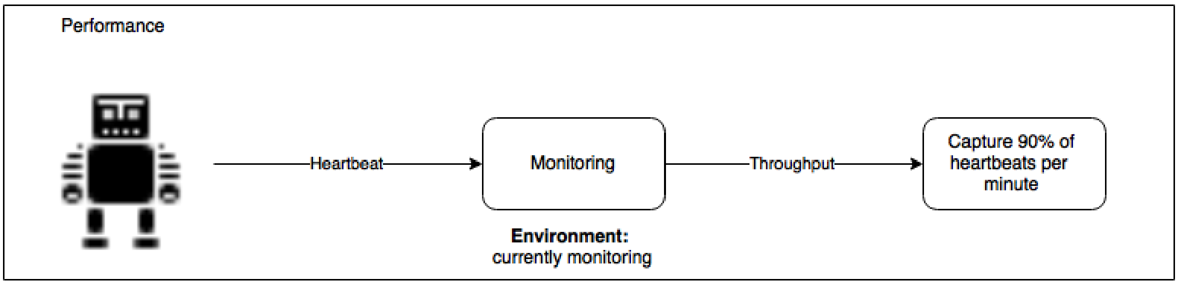


Figure 3.9.: Scenario 1: Performance

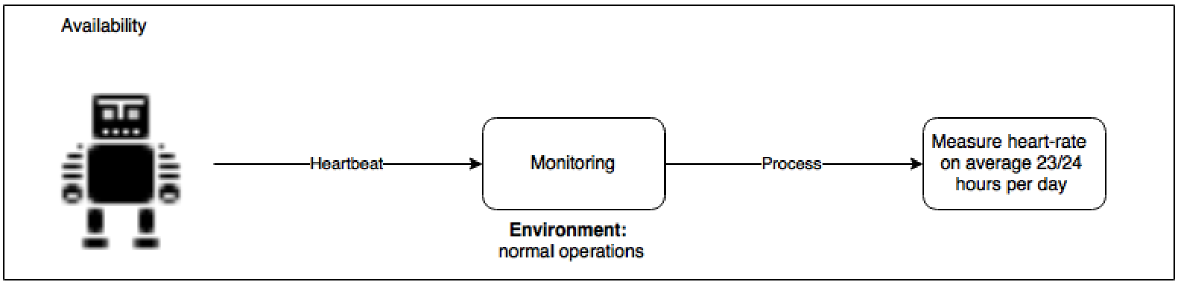


Figure 3.10.: Scenario 2: Availability

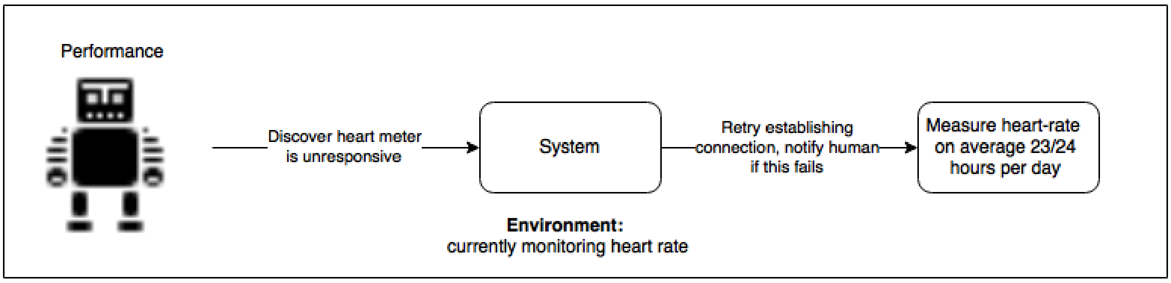

Performance
Discover heart meter is unresponsive
System
Retry establishing connection, notify human if this fails
Measure heart-rate on average 23/24 hours per day
Environment:
currently monitoring heart rate

Figure 3.11.: Scenario 3: Performance

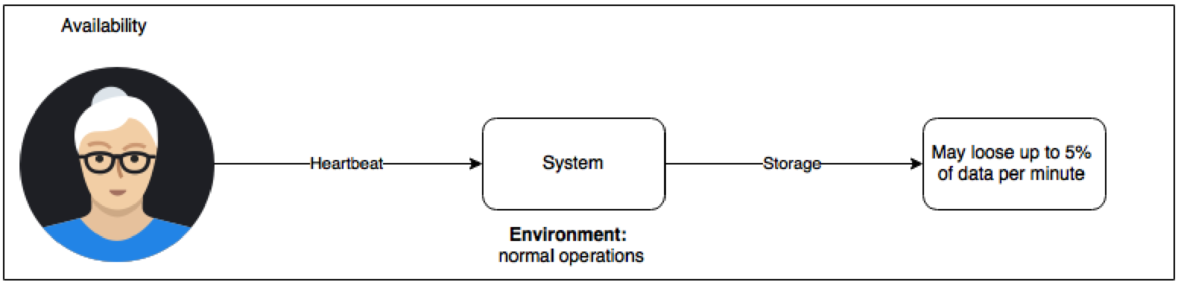


Figure 3.12.: Scenario 4: Availability

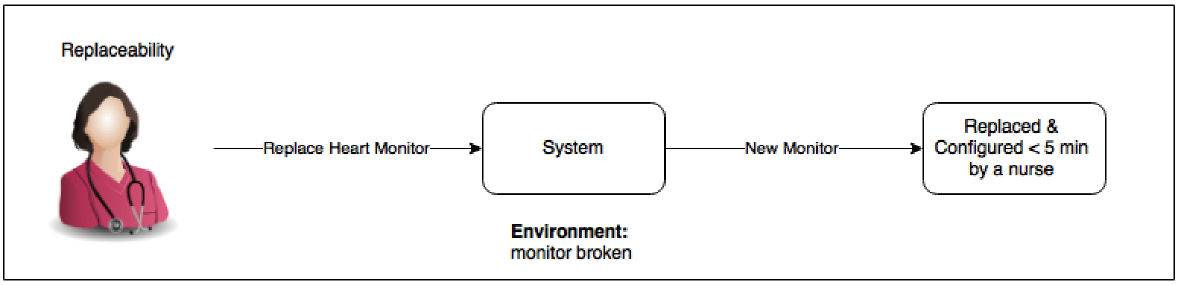


Figure 3.13.: Scenario 5: Replace-ability

Tactics

For an overview of de decisions consult 3.14. From the root system, the monitor, the scenario's hare handled which spawn certain choices, some of these generate new problems which have to be addressed, growing the solution tree.

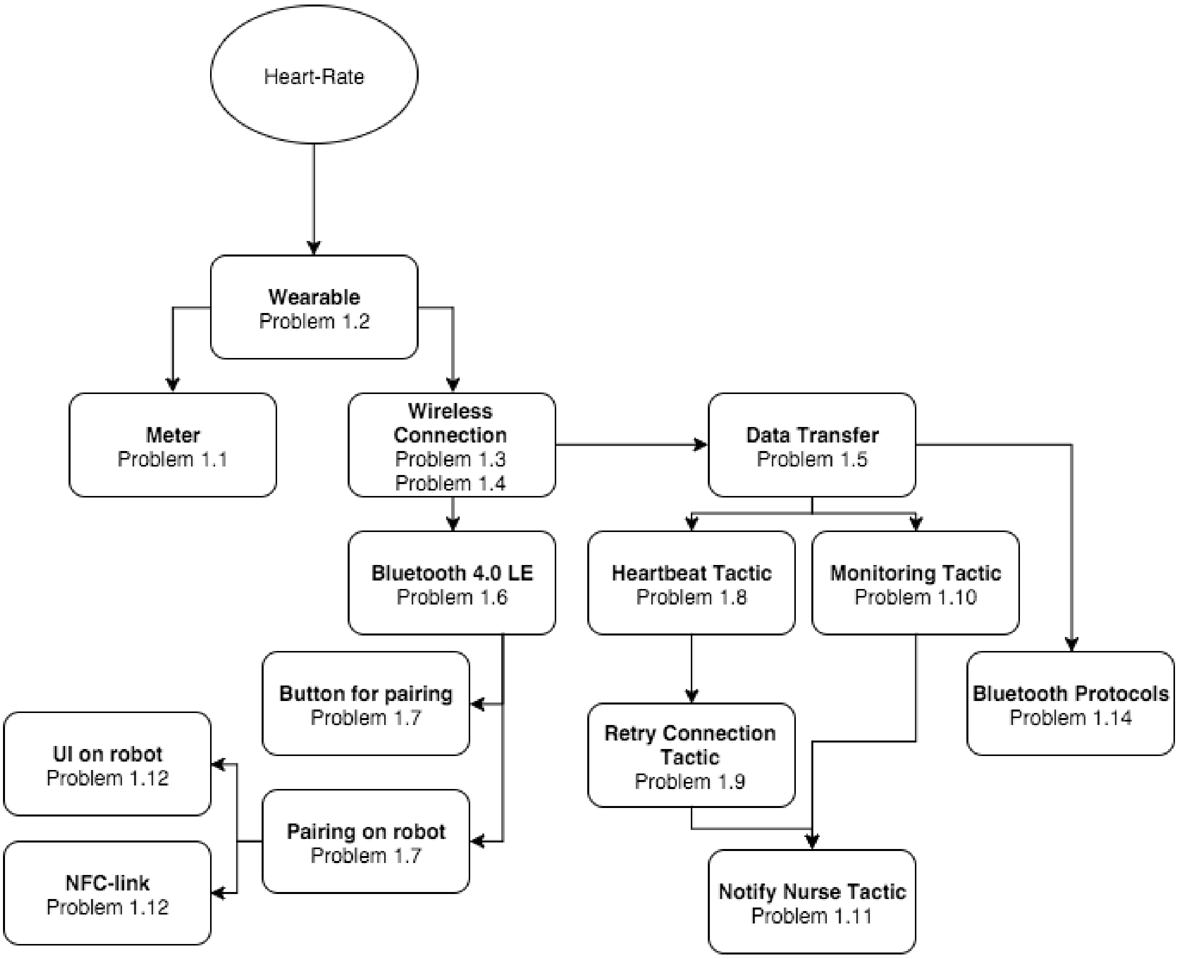


Figure 3.14.: Solutions overview

Below we will describe the tactics used to achieve the quality attributes of the heart rate monitor system.

We will start with 'Scenario 1: Performance' on page 28 as we deem this the most important. We will state the problem associated with the scenario and its solution with a rational. Furthermore we state the relations with other scenario?s and implications of this choice for other scenario?s which will be described in a similar fashion.

Table 3.1.: Problem 1.1: Capture 90% of heartbeats per minute.

Problem ID	1.1
Problem	Capture 90% of heartbeats per minute.
Solution	Use a quality heart-meter.
Assumption	Heart meter delivers the promised results. And fits in a wearable.
Related	‘Scenario 1: Performance’ on page 28
Rationale	When measuring the monitor should be accurate enough to capture 90% of heart beats. We wont go into detail about the actual meter itself (hardware component), we have found several meters on-line that are capable of measuring with great accuracy (95%) [20], this high accuracy is necessary to analyze heart-rates. Instead of developing our own hardware device it might actually be more feasible to buy an of the shelve solution for this if we can find something that matches our requirements.
Implications	The robot hardware must support the maximum data throughput. The theoretical maximum throughput of low energy bluetooth is a mere 236.7kbps [25] which is easy to handle.

Table 3.2.: Problem 1.2: (Hardware) Measure heart-rate on average 23 hours per day.

Problem ID	1.2
Problem	(Hardware) Measure heart-rate on average 23 hours per day.
Solution	Use a wearable for measuring heart-rate.
Assumption	N.A.
Related	‘Scenario 2: Availability’ on page 28 ‘Problem 1.8: (Software) Measure heart-rate on average 23 hours per day’ on page 35.
Rationale	By using a hardware module embedded within the robot we will not be able to fulfil the requirement of measuring a person for 23 hours a day because this would seriously hinder a persons mobility. To meet this requirement without interfering too much with a persons daily activities we will need to use a separated hardware component that can be worn by an elderly. This so called wearable will allow us to measure an elderlies heart-rate continuously and thus meet our requirement. We assume the wearable does not need to be continuously updated to keep functioning.
Implications	‘Problem 1.3: Connection to the Robot’ on page 32 ‘Problem 1.4: Comfort for the elderly’ on page 32 Pause on user request (e.g. when taking a shower) Updating the wearable for critical updates is done by a service engineer. Usability of the device.

Table 3.3.: Problem 1.3: Connection to the Robot

Problem ID	1.3
Problem	Connection to the Robot.
Solution	Wireless Connection between wearable and robot.
Assumption	Robot located in the same room as the elderly.
Related	'Problem 1.2: (Hardware) Measure heart-rate on average 23 hours per day' on page 31 'Problem 1.4: Comfort for the elderly' on page 32.
Rationale	Using a wired connection between the wearable and the robot would be the most easy and least complicated solution since it has less implications (no battery needed, no pairing needed). However this does not comply with 'Problem 1.4: Comfort for the elderly' on page 32. A wireless solution is needed, even though this has some negative implications (see implications).
Implications	'Problem 1.6: Battery-Life of the wearable' on page 34 'Problem 1.5: Data-Transfer' on page 33 'Problem 1.7: Pairing between wearable and robot' on page 34 'Problem 1.13: Security of the system' on page 37

Table 3.4.: Problem 1.4: Comfort for the elderly

Problem ID	1.4
Problem	Comfort for the elderly.
Solution	Wireless Connection between wearable and robot.
Assumption	Robot located in the same room as the elderly.
Related	'Problem 1.2: (Hardware) Measure heart-rate on average 23 hours per day' on page 31 'Problem 1.3: Connection to the Robot' on page 32.
Rationale	The most comfortable wearable (without looking at ergonomics) would be a device that does not limit the elderly in his or her daily activities. Using a solution that requires a cable is therefore not possible, it would simply limit a person in his or her activities. The best and only solution in this case is to use a wireless connection between the robot and the wearable device.
Implications	'Problem 1.6: Battery-Life of the wearable' on page 34 'Problem 1.5: Data-Transfer' on page 33 'Problem 1.7: Pairing between wearable and robot' on page 34.

Table 3.5.: Problem 1.5: Data-Transfer

Problem ID	1.5
Problem	Data-Transfer
Solution	Bluetooth 4.0 (LE) between wearable and robot, use HRP protocol for data.
Assumption	Robot located in the same room as the elderly. Bluetooth 4.0 LE will provide us with enough bandwidth. Protocol for transferring data will be HRP (Heart Rate Protocol) [22].
Related	'Problem 1.3: Connection to the Robot' on page 32 'Problem 1.4: Comfort for the elderly' on page 32 'Problem 1.6: Battery-Life of the wearable' on page 34.
Rationale	To transfer data between the wearable and the robot there are a couple of options such as (i) infrared, (ii) Wifi and (iii) Bluetooth 4.0 (LE). Infrared (i) is not an option as the sender needs to be in line of sight of the receiver, constraining the user in his movements. WiFi (ii) is a viable option but it is a rather resource heavy protocol, straining the battery live. Also the specifications are more than we need [16]. Bluetooth 4.0 LE (iii) is deemed the best option since it is cheaper and more energy efficient than WiFi (and regular Bluetooth 4.0), another benefit is that Bluetooth supports service discovery that can aid in pairing (see implication). Some other preferable specifications of Bluetooth LE are a 100m range and a low latency with a bandwidth of 1.0 Mb/s which should be sufficient for our data needs [15]. Bluetooth also offers a standard data-protocol for system status [21] and heart-rate monitors (HRP) [22]. This would ensure compatibility between the robot and wearables from possible different manufacturers. Choosing Bluetooth and all its features does negatively impact modifiability as the Bluetooth protocol can not simply be replaced if many other features depend on it, but as this is not a QA we can safely neglect this. Also the best viable alternative, WiFi, would have the same drawback.
Implications	'Problem 1.6: Battery-Life of the wearable' on page 34 'Problem 1.7: Pairing between wearable and robot' on page 34 Wearable and robot will need bluetooth. Software module responsible for storage and/or analytics must use the HRP protocol

Table 3.6.: Problem 1.6: Battery-Life of the wearable

Problem ID	1.6
Problem	Battery-Life of the wearable.
Solution	Bluetooth 4.0 (LE).
Assumption	Robot located in the same room as the elderly.
Related	‘Problem 1.3: Connection to the Robot’ on page 32 ‘Problem 1.4: Comfort for the elderly’ on page 32 ‘Problem 1.6: Battery-Life of the wearable’ on page 34
Rationale	Since a wearable device is used battery-life will be important, for a wireless connection this means we need a low energy solution. The most energy efficient networking technology today is Bluetooth 4.0 LE (Low Energy). This technology only uses 0.01/0.5watt compared to 1watt for normal Bluetooth 4.0 and 5watt for WiFi.[15][16]. Even though energy use is low with Bluetooth the battery or device must eventually be replaced, it goes outside of the scope to calculate battery-size but when looking around at wearable heart-rate monitors that use Bluetooth on the market it must be possible to guarantee a battery life of more than one day.
Implications	‘Problem 1.12: Replaced & Configured <5 min by a nurse’ on page 37

Table 3.7.: Problem 1.7: Pairing between wearable and robot

Problem ID	1.7
Problem	Pairing between wearable and robot.
Solution	Select the desired discovered Bluetooth wearable on the robot.
Assumption	Wearable has a physical button to enable Bluetooth service discovery. Wearable has a physical ID that matches the Bluetooth discovery ID.
Related	‘Problem 1.3: Connection to the Robot’ on page 32 ‘Problem 1.4: Comfort for the elderly’ on page 32 ‘Problem 1.5: Data-Transfer’ on page 33 ‘Problem 1.12: Replaced & Configured <5 min by a nurse’ on page 37
Rationale	By using Bluetooth service discovery a nurse can pair a wearable to the robot by simply pressing the pair button (enabling discovery) on the wearable. The robot will than be able to locate the wearable by simply scanning for Bluetooth devices. Big advantage of this Bluetooth discovery is that it is build-in with the Bluetooth specification and therefore compatible with all Bluetooth devices. Another option would be to use a RFID scanner, this would allow the nurse to press the pair button on the wearable and tap the wearable on the robot which could then read the wearable’s id from its NFC-tag and connect [23]. A UI in this scenario would still be necessary incase the tag is missing or malfunctioning. As this would still need a backup the overhead in cost and complexity is not worth it.
Implications	Robot needs interface to discover the wearable.

Table 3.8.: Problem 1.8: (Software) Measure heart-rate on average 23 hours per day.

Problem ID	1.8
Problem	(Software) Measure heart-rate on average 23 hours per day.
Solution	Robot detects fault in wearable by using the heartbeat fault detection mechanism.
Assumption	N.A.
Related	'Scenario 2: Availability' on page 28 'Problem 1.2: (Hardware) Measure heart-rate on average 23 hours per day' on page 31
Rationale	Since the wearable will send periodic messages with vital signs data to the robot this signal can be used for the heartbeat fault detection mechanism without the need to increase bandwidth usage and complexity by implementing a separate heartbeat mechanism on the wearable.
Implications	'Problem 1.9: Recover from detected fault in the wearable' on page 35

Table 3.9.: Problem 1.9: Recover from detected fault in the wearable.

Problem ID	1.9
Problem	Recover from detected fault in the wearable.
Solution	Retry connecting with the wearable.
Assumption	Robot and wearable were paired before connection loss.
Related	'Scenario 3: Performance' on page 28 'Problem 1.8: (Software) Measure heart-rate on average 23 hours per day' on page 35.
Rationale	Since the bluetooth protocol allows for reconnection without manual intervention this should be tried first before taking more extreme measures [15].
Implications	'Problem 1.11: Retry connection fails or other issue with the wearable has been detected' on page 36.

Table 3.10.: Problem 1.10: (Software) Measure heart-rate on average at least 23 hours per day.

Problem ID	1.10
Problem	(Software) Measure heart-rate on average 23 hours per day.
Solution	Robot detects issue with the wearable by monitoring the system state.
Assumption	The wearable broadcasts device-information.
Related	‘Scenario 3: Performance’ on page 28 ‘Problem 1.2: (Hardware) Measure heart-rate on average 23 hours per day’ on page 31.
Rationale	The wearable sends device-information to the robot whenever it transmits data, this data can be used for monitoring the wearable. To comply with ‘Scenario 2: Availability’ on page 28 the robot must be able to detect an empty battery to prevent a sudden loss of connection. This is actually easily done by using yet another of the standard Bluetooth protocols, for instance the battery service module [21].
Implications	‘Problem 1.11: Retry connection fails or other issue with the wearable has been detected’ on page 36.

Table 3.11.: Problem 1.11: Retry connection fails or other issue with the wearable has been detected.

Problem ID	1.11
Problem	Retry connection fails or other issue with the wearable has been detected.
Solution	Notify nurse.
Assumption	System for notifying nurse in place.
Related	‘Scenario 3: Performance’ on page 28 ‘Problem 1.8: (Software) Measure heart-rate on average 23 hours per day’ on page 35 ‘Problem 1.10: (Software) Measure heart-rate on average at least 23 hours per day’ on page 36
Rationale	If both recovery methods have failed a nurse must be called to restore the system to normal operations. The nurse must be called to reconnect or replace the wearable as all technical measures have been exhausted.
Implications	Remote server has a service for alerting a nurse.

Table 3.12.: Problem 1.12: Replaced & Configured <5 min by a nurse.

Problem ID	1.12
Problem	Replaced & Configured <5 min by a nurse.
Solution	Pairing between wearable and robot via Bluetooth with an UI.
Assumption	Nurse is familiar with the robot (recived training).
Related	‘Scenario 5: Replace-ability’ on page 29
Rationale	Replacing the heart-rate measurement monitor should be quick and simple for the nurse to comply with ‘Scenario 5: Replace-ability’ on page 29. Replacement is necessary in case of an empty battery or when the device is malfunctioning. Instead of allowing a nurse to replace (or recharge) a battery we have decided to let a nurse replace the device in its completeness, this would speedup replacement significantly since if one wants to replace a battery it will already be necessary to remove the device from the elder. This would also make it possible to choose a rechargeable wearable.
Implications	‘Problem 1.7: Pairing between wearable and robot’ on page 34

Table 3.13.: Problem 1.13: Security of the system.

Problem ID	1.13
Problem	Should the connection between the robot and the wearable be secure?
Solution	Bluetooth encryption.
Assumption	Nurse is familiar with the robot (received our training).
Related	‘Problem 1.3: Connection to the Robot’ on page 32
Rationale	The wearable will be transmitting patient data, this data is not considered to be sensitive. It is however a good practice to encrypt personal data when possible and therefore recommended to do so. Bluetooth however is always encrypted, its level of encryption is not very high but deemed more then enough for this application.
Implications	N.A.

Table 3.14.: Problem 1.14: Modifiability of the system.

Problem ID	1.14
Problem	Currently only heart-rates are supported, what if in the future we want to support blood pressure meters as well?
Solution	Support extra protocols.
Assumption	N.A.
Related	N.A.
Rationale	We already defined that for the connection between the wearable and the robot we will use bluetooth 4.0 LE, an advantage of this setup is that we get access to multiple data protocols which are all build-in [21]. For the heart-rate monitor we decided to use the standard HRP (Heart Rate Protocol) [22], for other vital monitors similar protocols exist including one for blood pressure. So in case we want to support more types of measurements we will only need to update the software component that analyses the data. Using the build-in bluetooth protocols would make developing this software easier and more testable. Most implications will therefore be with this analytics module which should probably be designed in a modular way (e.g. pipe-filter analysing the data).
Implications	Analysing software must be easily modifiable.

3.4. System 2: Cardiac arrest alarm

In depth description of the system that will sound an alarm when a cardiac arrest has been detected. For an overview of the system see figure: 3.21

Scenarios

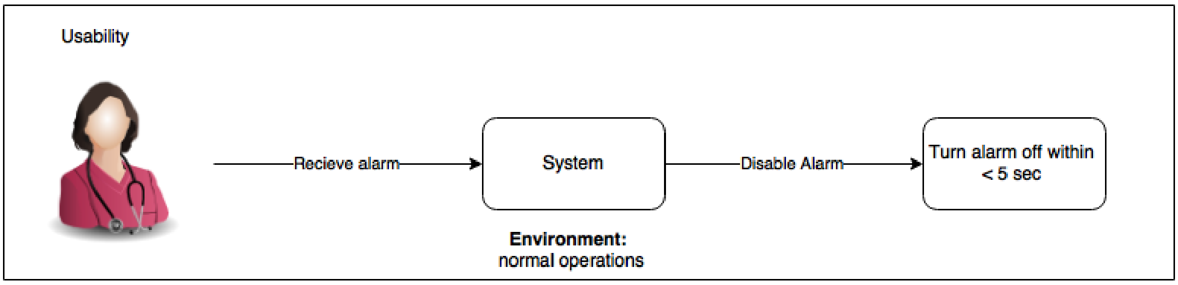


Figure 3.15.: Scenario 1: Usability

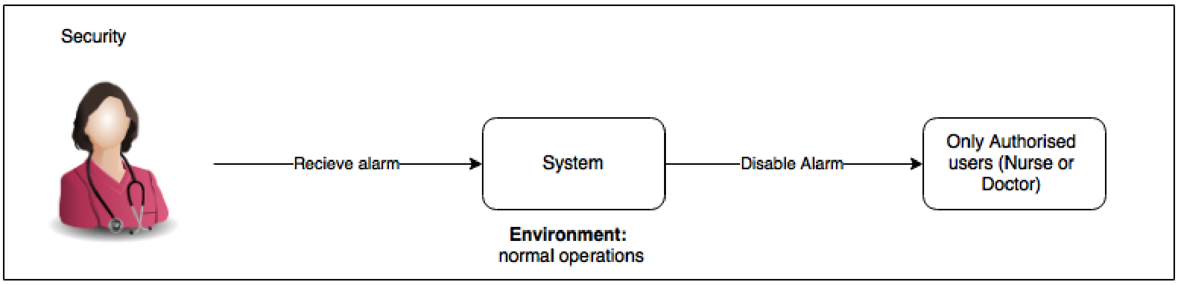


Figure 3.16.: Scenario 2: Security

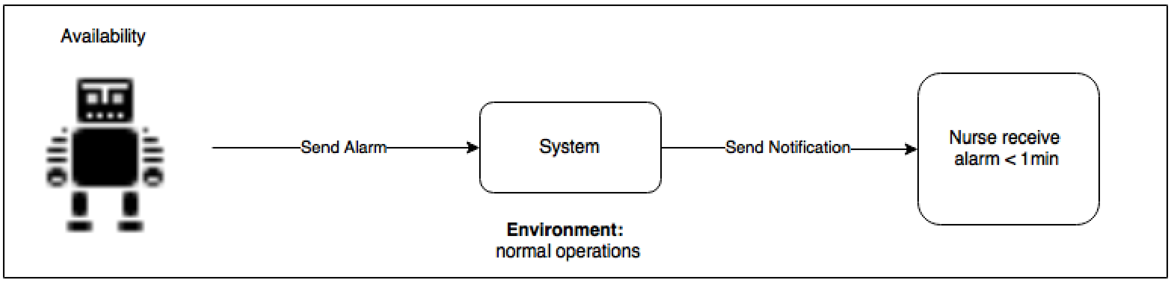


Figure 3.17.: Scenario 3: Availability

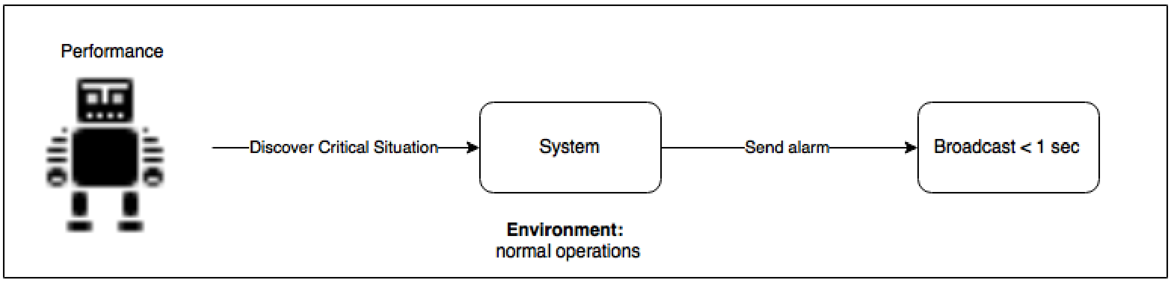


Figure 3.18.: Scenario 4: Performance

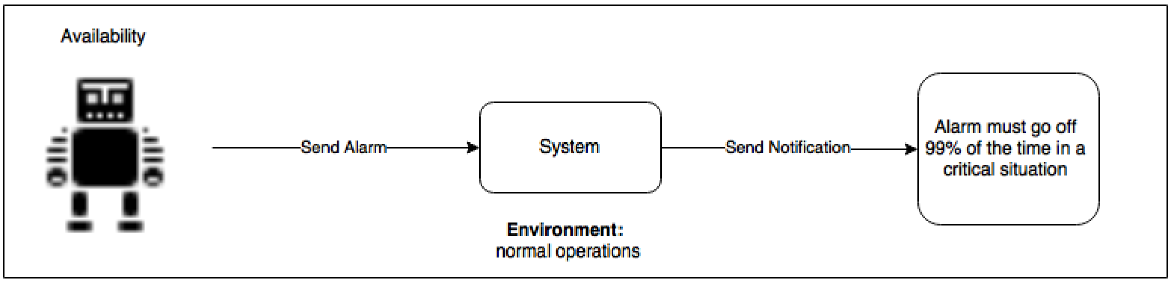


Figure 3.19.: Scenario 5: Availability

Tactics

For an overview, consult Figure 3.20. From the root system, the alarm, the scenario's are handled which spawn certain choices, some of these generate new problems which have to be addressed, growing the solution tree.

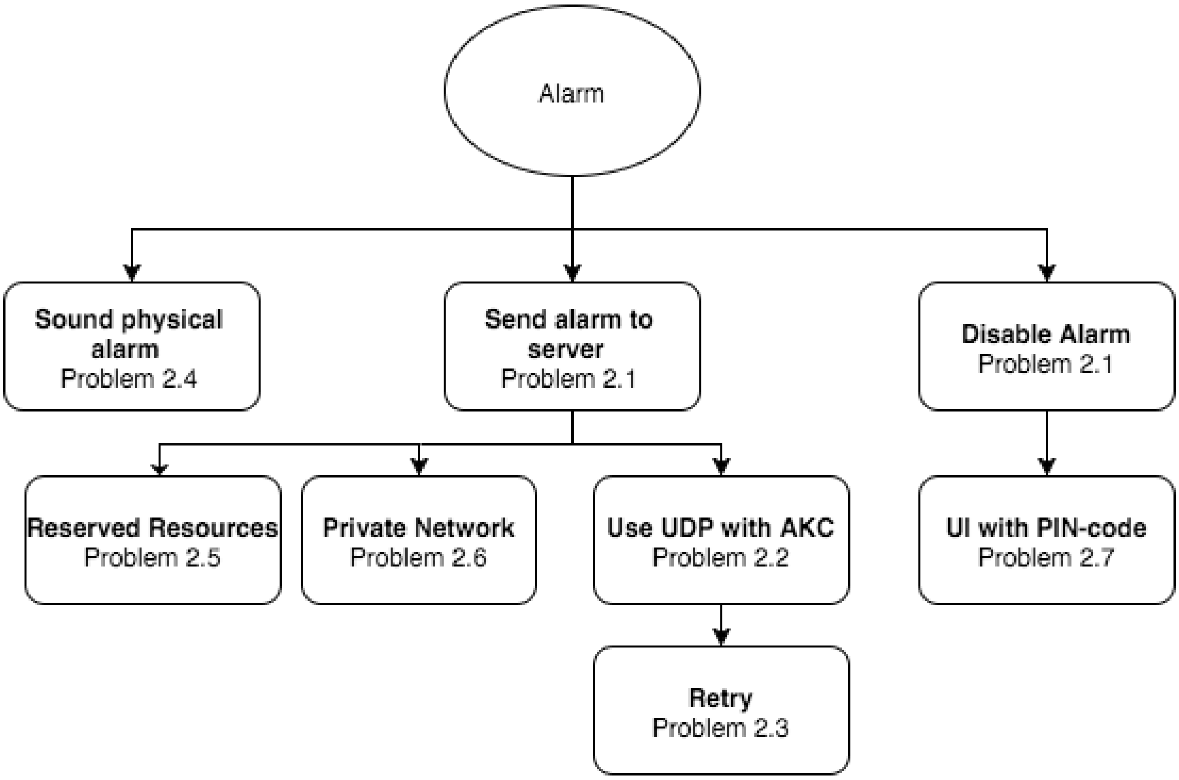


Figure 3.20.: Solutions overview

Below we will describe the tactics used to achieve the quality attributes of the cardiac arrest alarm system.

We will start with 'Scenario 3: Availability' on page 39 as we deem this the most important. We will state the problem associated with the scenario and its solution with a rational. Furthermore we state the relations with other scenario's and implications of this choice for other scenario's which will be described in a similar fashion.

Table 3.15.: Problem 2.1: Notify nurse <1 min of cardiac arrest.

Problem ID	2.1
Problem	Nurse must be notified within 1 minute of detecting a cardiac arrest.
Solution	Send alarm message to server (which sends it to the nurse).
Assumption	N.A.
Related	'Scenario 3: Availability' on page 39
Rationale	Other possibilities would be that the Quality Care Robot directly notifies a nurse. The implication of this would be that every robot must know which nurse is assigned to which patient at any given time and how to notify this person. This data should then be updated on each robot to stay in sync with the real situation from a central point. Implications of this concerns data concurrency across all robots, data security and modifiability of the notification mechanism. Given that this central point would already have this information it makes more sense for every robot to delegate the notifying to this central point. Now there is only one point to secure and modify, also this keeps the robot simpler and the server could choose the method to propagate the message.
Implications	'Problem 2.5: Broadcast alarm <1 sec' on page 44, 'Problem 2.6: Only sound alarm from a valid source' on page 44, System for notifying the nurse on the server, security and modifiability of said system.

Table 3.16.: Problem 2.2: Send alarm to server with average latency of 1 sec.

Problem ID	2.2
Problem	Send alarm to server with average latency of 1 sec.
Solution	Use UDP with acknowledgments from robot to server.
Assumption	Robot knows where to reach the server.
Related	'Scenario 4: Performance' on page 40, 'Problem 2.1: Notify nurse <1 min of cardiac arrest' on page 42
Rationale	TCP ensures ordering, integrity and reliability of send packages but at a possible performance penalty. Whereas UDP makes no such insurances but is generally faster [24]. As performance is important for this quality attribute the alarm messages will be send over UDP to the server. An implication of this is dealing with the unreliability of the UDP protocol (i.e. packages can be corrupt or not arrive at all), this will be resolved by using acknowledgments. As long as the server does not acknowledge that it received the alarm message the robot will keep retrying (see implications). This does imply more implementation work compared to TCP.
Implications	'Problem 2.3: No alarm ACK received from server' on page 43, 'Problem 2.4: No connection to server' on page 43, The server must reply with an acknowledgment upon receiving the alarm.

Table 3.17.: Problem 2.3: No alarm ACK received from server.

Problem ID	2.3
Problem	No alarm ACK received in 50 ms (Retry).
Solution	Retry until an acknowledgment is received or the alarm on the robot is turned off.
Assumption	N.A.
Related	‘Scenario 3: Availability’ on page 39, ‘Problem 2.2: Send alarm to server with average latency of 1 sec’ on page 42
Rationale	The retry delays are based on Apache ActiveMQ [17]. We use the standard delays, with a bigger initial delay to account for variability in the network connection. The retry scheme is based on the principle that after each retry the chance for success decreases because the chance that there is a serious network issue that prevents the messages to be delivered increases. Therefore the delay with which every retry is send is doubled. Whereas simply retrying to send the message every 50ms would basically be a light DOS attack on your own server.
Implications	N.A.

Table 3.18.: Problem 2.4: No connection to server.

Problem ID	2.4
Problem	No connection to server.
Solution	Use physical speaker to sound alarm.
Assumption	N.A.
Related	‘Scenario 3: Availability’ on page 39, ‘Problem 2.2: Send alarm to server with average latency of 1 sec’ on page 42
Rationale	To increase the chances of complying with ‘Scenario 3: Availability’ on page 39, also in a situation where there is no connection to the server, the physical speaker of the robot will be sounding an alarm whenever there is a critical health situation detected. This could potentially alarm any bystanders or personnel roaming the hallways. Another possibility would be to choose for a fault detection tactic to discover if there is a fault in the connection between the robot and the server. For instance by using ping/echo, the server could then notify a service engineer who would be tasked to fix the problem. However, the chance of having no connection while at the same time there occurs a cardiac arrest is minimal. We choose to accept this chance based on the threshold set by ‘Scenario 3: Availability’ on page 39. Also if there would be a prolonged connection error this would be notified fairly quickly by the user or one of the nurses when interacting with the robot because many features depend on this connection.
Implications	N.A.

Table 3.19.: Problem 2.5: Broadcast alarm <1 sec.

Problem ID	2.5
Problem	Broadcast alarm <1 sec.
Solution	Reserve enough server resources so at least one alarm can be processed at any given time.
Assumption	N.A.
Related	'Scenario 3: Availability' on page 39, 'Problem 2.1: Notify nurse <1 min of cardiac arrest' on page 42
Rationale	We require the alarm event to be forwarded to the correct nurse without delay. If we would reserve server resources, such that at any given time those resources can be used to process the alarm, we do not introduce processing delays in the server. Another option would be to use a priority queue. The alarm event would be assigned the greatest priority. However, when the server is fully occupied, the alarm event will receive first priority in the queue, but still has to wait until a currently running process is completed.
Implications	Less resources available on the server for other tasks, but this is minimal.

Table 3.20.: Problem 2.6: Only sound alarm from a valid source.

Problem ID	2.6
Problem	Only sound alarm from a valid source.
Solution	Sufficiently solved by having the server only accept alarms from within the nursery home.
Assumption	Robot and server are located within the nursery home.
Related	'Problem 2.1: Notify nurse <1 min of cardiac arrest' on page 42
Rationale	Invalid or unauthorized sources (i.e. hackers) should not be able to easily sound an alarm. If the source that sounds the alarm (the robot) would be located on an outside network, i.e. the Internet, it would be needed to encrypt the alarm data and validate the source to prevent just anybody from sending an alarm message to the server. But because the server and robot are within the same intra-net in the nursery home, the server can simply only allow alarm messages that originate from within this network. The hacker would also need to know the form of this message, this is usually done by intercepting an earlier message of this type, but because these alarm messages would be send only very occasionally the chance of all this happening and forming real issues is negligible.
Implications	N.A.

Table 3.21.: Problem 2.7: Only authorised users (nurse, doctor) can disable the alarm and must be able to do so within 5 seconds.

Problem ID	2.7
Problem	Only authorized users (nurse, doctor) can disable the alarm and must be able to do so within 5 seconds.
Solution	Disable the alarm with a 4 digit code through the robots GUI
Assumption	N.A.
Related	Scenario 3.16
Rationale	When a caretaker enters the room of a person in need of assistance he or she should be able to disable the alarm quickly. The best solution would be to provide the robot with a physical button for deactivation. However this does not comply with 'Scenario 2: Security' on page 39, which states that only authorized personnel can turn off the alarm. Another solution to this could be applying an NFC reader to the robot. With NFC a caretaker can simply tap the robot with his/her ID-card to disable the alarm, implications however are added complexity and higher cost for the robot. An other disadvantage of using this technology is that the robot needs to be able to verify a caretaker with the server, meaning that when there is no connection the alarm cannot be turned off. Also when picking NFC there must always be a backup method in case a caretaker forgets his/her ID-card. To solve this implication a user-interface on the robot could be build that accepts a four digit code know to the caretakers. We decided that in this phase of the project we will only use a GUI to deactivate the alarm, to keep the robot as simple as we can. It would however still be possible to implement NFC at a later stage since NFC would also require a GUI as backup.
Implications	Robot needs a GUI that when the alarm is activated displays a deactivation page with a four-digit pin. Server must be able to push a new code to the robots.

3.5. System overview

Figure 3.21 gives an overview of the system that encompasses the tactics used to comply with the QA's as described in the previous chapters.

- Specification on how the *monitoring module* calculates critical situations, or decides which data to send to the server is not yet modeled.
- The server resources being reserved for sending notifications to nurses is in fact a very small portion of the total server resources (whereas Figure 3.21 might suggest otherwise).
- The hardware device on which the nurse receives the notifications is not modeled, since it is not a part of our system. It should however be taken into consideration when looking at the diagram. We have assumed this to be some kind of mobile device.
- The *wearable module* can be replaced with a simular one without affecting the system.

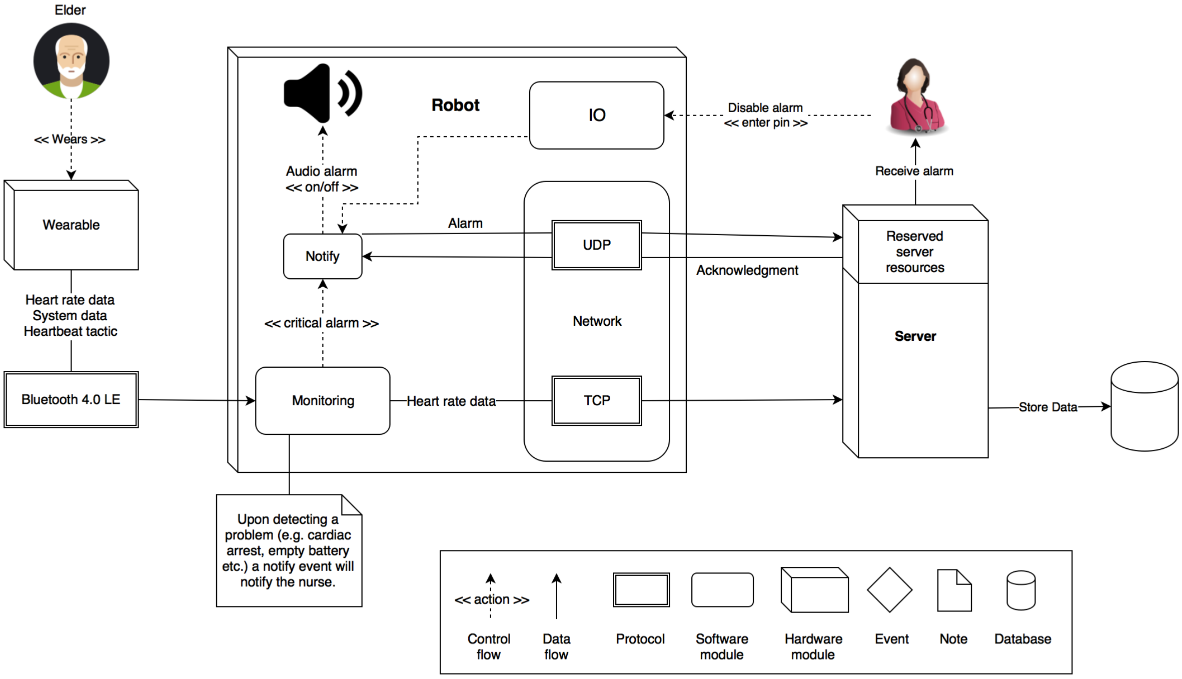


Figure 3.21.: System overview

4. References

[1] Jan Bosch, *Software Architecture: The Next Step*, University of Groningen, Department of Computing Science

[2] Chandan Datta, and Hong Yul Yang, I-Han Kuo, Elizabeth Broadbent and Bruce A MacDonald, *Software platform design for personal service robots in healthcare*, 2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)

[3] Chandimal Jayawardena, Nilufar Baghaei, Kathiravelu Ganeshan and Abdolhossein Sarrafzadeh, *Designing a Socially Assistive Companion Robotic Wheel Chair: RoboChair*, 2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)

[4] Min Yang Jung and Peter Kazanzides, *Run-time Safety Framework for Component-based Medical Robots*, Department of Computer Science, Johns Hopkins University

[5] Carlos Busso, Zhigang Deng *, Serdar Yildirim, Murtaza Bulut, Chul Min Lee, Abe Kazemzadeh, Sungbok Lee, Ulrich Neumann* , Shrikanth Narayanan, *Analysis of Emotion Recognition using Facial Expressions, Speech and Multimodal Information* , Emotion Research Group, Speech Analysis and Interpretation Lab Integrated Media Systems Center, Department of Electrical Engineering, * Department of Computer Science Viterbi School of Engineering, University of Southern California, Los Angeles

[6] Vincent Driessen, A successful Git branching model,
`http://nvie.com/posts/a-successful-git-branching-model/`

[7] ROS, Robotic Operating System
`http://www.ros.org/`

[8] OpenStack
`http://www.openstack.org`

[9] Erico Guizzo Cloud Robotics: Connected to the Cloud, Robots Get Smarter
`http://spectrum.ieee.org/automaton/robotics/robotics-software/cloud-robotics`

[10] Florian Johannssen NAO in the cloud
`http://www.slideshare.net/FloJo24/nao-in-the-cloud-talk`

[11] RoboEarth
`http://roboearth.org/cloud_robotics/`

[12] Moscov Model
`http://www.dsdm.org/content/10-moscow-prioritisation`

[13] Philippe Kruchten, *Architectural Blueprints The 4+1 View Model of Software Architecture*, Rational Software Corp. 638-650 West 41st Avenue Vancouver, B.C., V5Z 2M9 Canada

[14] Mobile Medical Applications, Guidance for Industry and Food and Drug Administration Staff, V.B p.15
`http://www.fda.gov/downloads/MedicalDevices/.../UCM263366.pdf/`

[15] BLUETOOTH SPECIFICATION Version 4.0 [Vol 0]
`https://d3oygwokdsrfp4.cloudfront.net/f/0/0/57/76971b11cfadb54092b7793f5bf5a37c/o.pdf`

[16] 802.11ac: The Fifth Generation of WiFi, Technical White Paper
`http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-3600-series/white_paper_c11-713103.pdf`

[17] ActiveMQ: The Failover Transport. `http://activemq.apache.org/failover-transport-reference.html`

[18] MedicinePlus - Pulse
`https://www.nlm.nih.gov/medlineplus/ency/article/003399.htm`

[19] Survive Cardiac Arrest
`http://depts.washington.edu/survive/survival-formula.php`

[20] Best heart rate monitor and HRM watches
`http://www.wareable.com/fitness-trackers/best-heart-rate-monitor-and-watches`

[21] Bluetooth services
`https://developer.bluetooth.org/gatt/services/Pages/ServicesHome.aspx`

[22] Heart Rate Profile - Bluetooth
`http://developer.bluetooth.org/technologyoverview/documents/hrp_spec.pdf`

[23] NFC-Bluetooth
`https://gigaom.com/2013/08/08/still-not-a-wallet-nfc-has-a-second-life-as-a-safe-simpl`

[24] TCP/IP & UDP Protocols FAQs, Berk-Tek
`http://www.nexans.us/US/files/TCP_UDP_FAQ01G14.pdf`

[25] Gomez C, Oller J, Paradells J. Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. Sensors (Basel, Switzerland). 2012;12(9):11734-11753. doi:10.3390/s120911734.

A. Reference Architectures

The following architectural documents are useful as a reference when designing the architecture of the Quality Care Robot.

A.1. 2013 RAM Conference

At the *IEEE Conference of Robotics, Automation and Mechatronics (RAM), 2013 6th* the following paper was presented: *Software platform design for personal service robots in health care* [2].

The paper describes the software architecture design of a personal service robot which is to be applied in health care scenarios, see Figure A.1. This paper is especially useful as it describes the implementation of certain features into the framework that we are discussing in this document such as assistance with medication management and engaging with the system trough conversations. It also contains software and hardware layouts. It is a starting point for our architecture.

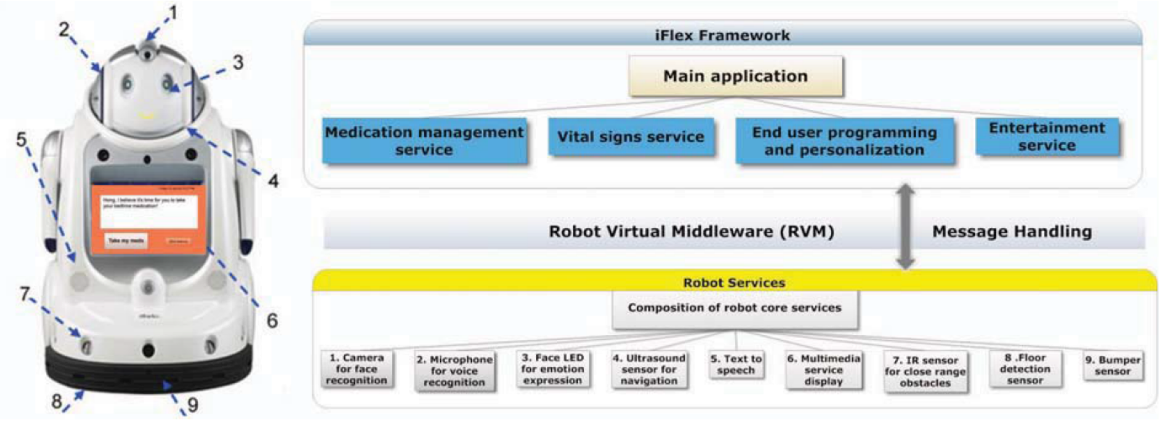

1
2
3
4
5
6
7
8
9
iFlex Framework
Main application
Medication management service
Vital signs service
End user programming and personalization
Entertainment service
Robot Virtual Middleware (RVM)
Message Handling
Robot Services
Composition of robot core services
1. Camera for face recognition
2. Microphone for voice recognition
3. Face LED for emotion expression
4. Ultrasound sensor for navigation
5. Text to speech
6. Multimedia service display
7. IR sensor for close range obstacles
8 .Floor detection sensor
9. Bumper sensor

Figure A.1.: iFLex architecture

The *RoboChair* [3] paper, presented aswell at the RAM conference, describes the design of a robotic wheelchair which is a socially assistive companion robot equipped with several medical measuring devices, See Figure A.2 It is particularly useful for us as it describes the combination of a voice controlled social companion robot combined with clinical measurement tools used for health predication and care assistance.

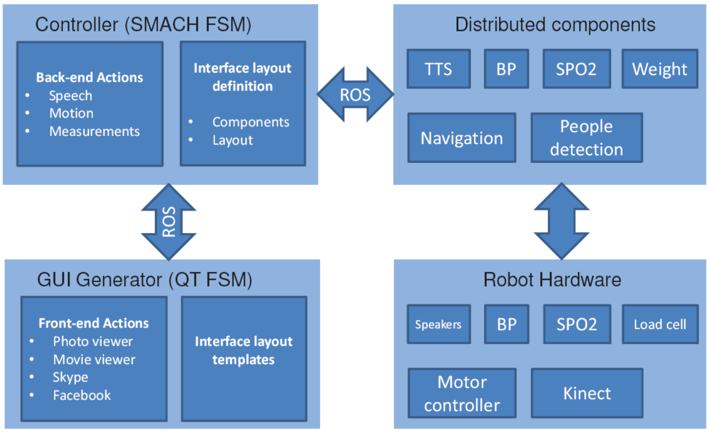


Figure A.2.: RoboChair architecture

A.2. Safety Framework

The safety framework architecture presented in the *Jung and Kazanzides* [4] paper describes the use of a robust safety framework with component based medical robots, see Figure A.3. This could be useful for our architectural design because of the way they combined the safety framework with the component layered design of the architecture. The architecture they described is also very important for Quality Care Robot as it will be build up of different components and needs to comply to strict safety measures with regards to its medical assistance features.

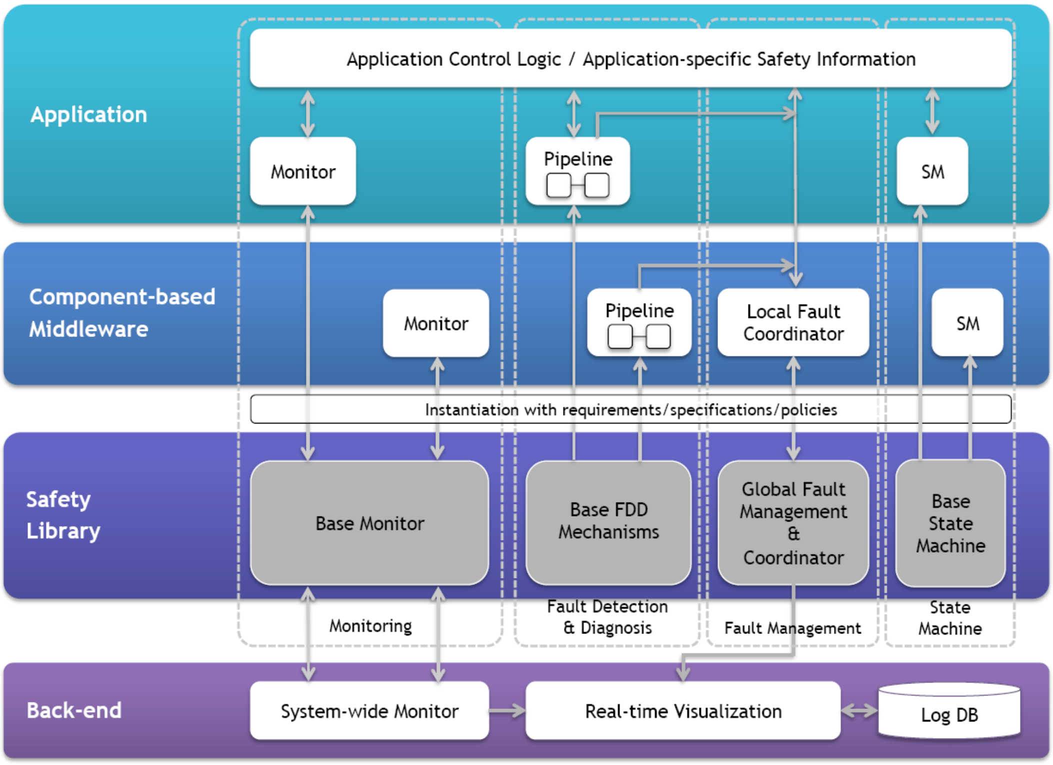


Figure A.3.: Safety framework

B. Reflection on Pipe Filters implementation

In hindsight, upon starting with the implementation of the pipe filter pattern, there were a few assumptions that we had made. The first and foremost being that the filter would only be used within the Quality Care software stack. Another one being that the pipeline would always run continuously, the result of this is that there is no good way to let the pipe stop only after all input has been processed.

To summarize, the following things could be improved:

- A compiled jar of the component should be easily extracted.
- Maybe supply a compiled jar.
- When implementing a design pattern as a loosely coupled component, be aware of other implementation scenario's than your own.
- Document how the component should be integrated within a software project.
- Create more/better examples or tests.

The following things went well:

- Easily understandable code documentation (for developers).
- With Maven a jar of only the pipe filter component is easily created.
- Used Interfaces and Abstract classes such that it is really easy to setup a new filter.

Our implementation of the pipe filter pattern using the code of `PT_W02_1` can be found here `https://github.com/tomvanduist/software-architecture/tree/Week_4_v1.1`

C. Architectural Patterns and Styles

This chapter describes a few architectural patterns that have to be used when building the Quality Care Robot. We describe why a certain patterns is chosen, why it fits the design (so far), and what the advantages are of choosing such a pattern.

C.1. Client-Server

The Quality Care Robot consists of two *or more separated* separate systems being the physical robot*(s)* itself, and the central processing unit. The former acts as a client. The client sends data for processing requests to the server. A processing request consists of data as input (e.g. audio file), and a query. For example: Mrs Janssen speaks to the Quality Care Robot; "Is it time for my pills already?". The audio file is send to the server for processing. The server (central processor unit) hosts and executes a number of computational-intensive applications and processes. These include analyzing speech and comprehending text (to some extent). The server returns a response, which is received by the robot. The robot translates the response back to an audio file and outputs the line; "At ten-thirty you will get your pills, handsome!".

The architectural style where the server is mainly used to execute (intensive) applications is called *application servers.* Added benefits of adopting the application servers pattern are *security*, *centralized data access*, and *ease of maintenance.* Since the data of all patients is stored in a centralized storage, it is easy for the different robots to access that data. We can build profound security into the server. The robot itself will not hold any private data besides some data from the heart and blood pressure meter. This data is necessary to do some real time processing in case of an emergency, this data will only be saved for one minute. The pill schedule is embedded into the pill dispenser (cartridge) if possible the validity will always be checked with the server, in case there is no network available it will follow the cartridges data. The final advantage - ease of maintenance - becomes apparent when servicing the client. If we have multiple servers running concurrently, the robots will be served by a backup server while the default server is serviced. The client is unaware of the maintenance, and operates normally.

For home automation we use and adjust the pattern *Client-Queue-Client.* This pattern lets two clients communicate to each other via a server. The server simply propagates data from one client to another. For example; Mrs Janssen wants to open the door. She says to the robot "Please open the door". This request is send to the server, which finds the right door-opening device, i.e. the door of Mrs Janssen. The door-opening device receives the response from the server and executes this response, i.e. the door opens. The other way around is possible as well. A relative of Mrs Janssen presses the doorbell (linked to the door-opening device). A request is send to the server to open the door. The request is propagated to the robot of Mrs Janssen, who says "Can I open the door?". The server interprets the response of Mrs Janssen, and either does or doesn't open the door. In a true Client-Queue-Client pattern the server does no computations. In our case we need the server to link the doorID to a robotID and vice versa. The advantages are:

- *Easier configuration management*: You only have to link a user to a door.

- *Replaceable*: any device will do, simply update the user information.

- *Extensible*: two users can live together and use the same door. Both users have the same device ID assigned.

- *Independent*: home automation devices are independent of the robot and other devices. They only communicate with the server. This reduces complexity when designing the robot. The functionality of a home automation device should just be accessible by a voice command for instance.

C.2. Model, View, Controller (MVC)

The MVC pattern has a layered architectural style. The MVC pattern is very simple, since it has only 3 layers. We can apply the MVC pattern to the robot. Note that we do not apply it to the central processor unit, since it lacks an UI. The View is the UI of the robot, see Figure 3.1 for a closer look at the UI. The Model consists of the DB inside the robot and the pill dispenser. Since we use a special cartridge for the pill dispenser - with a schedule programmed into the pill dispense - part of the data resides here. The controller consists of the Medical Management, which monitors medical data, and the QR Controller, which delegates input and output requests.

- *Testability*: Since one layer only uses itself and the layer below, it is easy to test separate layers. A tester can start testing the bottom layer for instance, without needing the higher layers. Since the interfaces between layers are well-defined, a tester can also use a part of the system that is already tested to test his new layer.

- *Manageability*: The dependencies are clear in a layered structure, which can be directly encoded in MAVEN. The developers do not have to be concerned about dependencies anymore, which makes it more manageable to write the different parts of the system.

- *Isolation*: When a blood pressure meter needs to be replaced, the MVC model assures us we can simply replace this part without breaking other parts of the system. A different blood pressure meter does not affect functionality of the UI for instance. Only the layer in which the blood pressure meter is active may need some adjustments.

C.3. Pipes and Filters

The pipes and filters pattern is mainly used for logging in our system. Large text files are produced when the robot is active. Vital signs, system status, conversation lines, etc. When something goes wrong, on-site maintenance (mechanic) wants to find the relevant information as fast as possible to fix the problem. When a number of filters are defined, the mechanic can simply link a number of them together to view the right information. The mechanic might want to access a conversation between Mrs Janssen from yesterday-evening. The robot called Mrs Janssen "A cranky old bag of bones"! The mechanic can insert a TimeFilter, a ContainsStringFilter, etc. to find what went wrong.

C.4. Shared-Data

See Figure C.1 for an overview. The top three processes are external processes which are used by the central processor unit. The bottom two processes are build by the Quality Care team. There is another process which makes sure that drug information is kept up to date, by reading the data from the external database, and updating the data in the shared data database. Concerns for the latter are security. The shared database needs to have extensive security measures. Otherwise the involved authorities might not feel confident in sharing their patient information. This pattern is chosen to reduce the complexity of the architectural infrastructure on the client-side of the application. Another reason for having a shared-data storage is availability. Since we want to connect users to each other via meetings we need to have easy access to all their data, in order to make the best matches when its relevant.

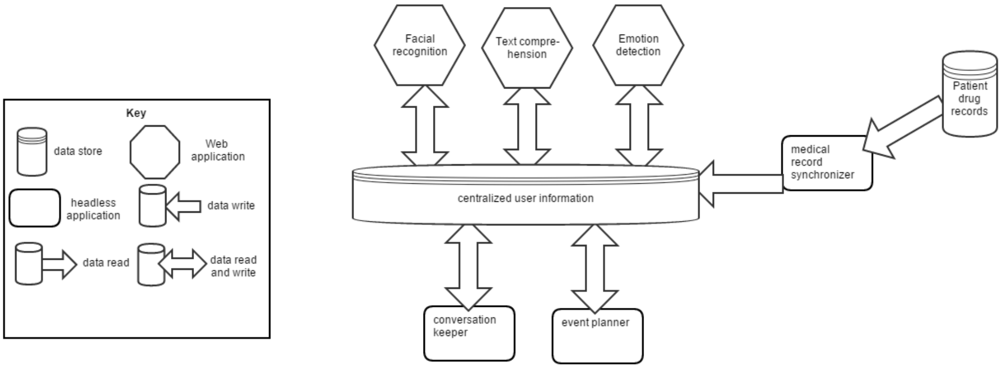


Figure C.1.: Shared-data model for server-side.

D. Review

D.1. Review goal

We like the review to focus on the part that addresses the Quality Attributes assignments. Specifically chapters 3.2, 3.3, 3.4 and 3.5. The reasons why the stakeholders want to detect for instance a cardiac arrest is not really important, we are instead interested in feedback with regards to the reasoning and anything that might be missing in chapter 3.3 and 3.4 and the system in chapter 3.5. Are design decisions sufficiently clear? Non-contradicting? Are there better decisions for problems we did not thought of?

Finally, is it clear what trade offs we made, where, and why we chose a certain QA over another?

D.2. Recieved feedback

We received the following feedback from group PT 1 that reviewed our document using the review goal stated above. For every comment we provided an answer to their question/comment (*Awnser*).

D.2.1. System 1

Problem 1.2: (Hardware) Measure heart-rate on average at least 23 hours per day. How do you make sure that the elderly wears and is able to wear the wearable 23 hours a day?

- Is it waterproof?
- Does the robot remind them to wear the device if they dont? (If yes, how are you going to develop that since youre going to use an existing wearable made by other developers)
- Can you detect if users are wearing the wearable? How do you differentiate between a user with no heart beat, and a wearable that is not being worn?

Awnser: We did not have a hard requirement for being waterproof, we decide to add an implication to our system that the user must be able to pause monitoring for a small duration of time. The robot does not remind the user, the nurse applies the wearable and when it stops detecting the nurse will be notified. If a user is not wearing the wearable no data will be collected, and thus an alarm will sound.

Problem 1.3: Connection to the Robot

- You stated that youre buying an off the shelf solution instead of developing the wearable yourself. Doesnt this affect maintainability? The wearable may be working with the robot properly now, but what if the company who makes those wearables stops maintaining your version of the wearable.

- what if it rolls out an update which is not compatible with the robot?
- Does the wearable need firmware updates? Who updates them? Who tests the integration with the robot? Problem 1.4: Comfort for the elderly
- What if the user is allergic for the material of the wearable? Are you going to use different materials for those users? Problem 1.5: Data transfer
- How often do you (try to) synchronize the wearable with the robot? Or do users only do this manually by pressing the pair button? (Do you trust users with this?)

Awnser: Off the shelve will not affect maintainability because we are using the standard blue-tooth protocol. This means that even if we choose a different vendor the connection will be standardized. For updates we decided to let our service engeneer handle this, if an update from our vendor is approved he/she can update the wearable devices (we added an implication for this to the system). We did not consider allergies for the user, we think this could be easily solved by using different materials or a non-allergenic material. Because this will not influence our decisions we decided not to include this in our reasoning. Syncing is also specified by the blue-tooth protocol, the user does not do anything, the robot simply makes a request to clear the wearables buffer.

Problem 1.12: Replaced & Configured <5 min by a nurse:

- If the nurse replaces a wearable with empty battery with another wearable, won't the wearable attempt to pair with another robot (since your robots are very close to each other in the nursing home)? Does the nurse need to reset the pairing? Problem 1.14:
- What if Bluetooth 4.0 is considered even less safe or less battery efficient than newer
- data transfer standards in the near future?

Awnser: Bluetooth pairing will forget the previous device when pairing with a new one. We will not consider newer technologies, because where will you draw the line?

Overall:

- What about the testability?
- Will you run regression/integration tests?
- How much code coverage?
- What about the maintainability of the code?

Awnser: We thought about these things but chose to focus on different parts of the system and work those out in depth. Things like code coverage will be hard to determine and depend on the methods responsibilities, regression and integration tests are of-course necessary. We do believe our testability is high because we are using a standard protocol and modular components that can also be tested independently from each other

D.2.2. System 2

Scenario: A patient experiences cardiac arrest. The nurse that is notified is on holiday / on the toilet / otherwise not noticing the alert The nurse does not disable the alarm and the patient dies.

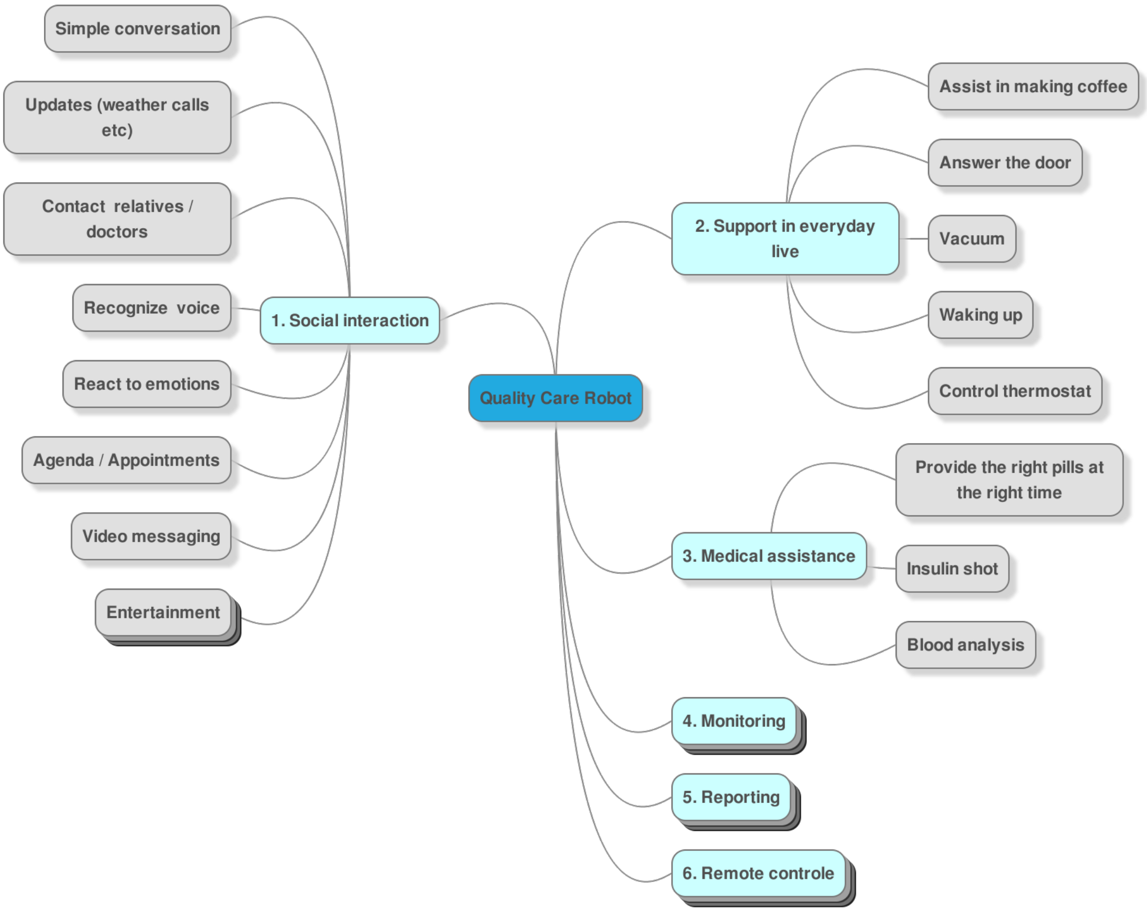
- How do you deal with a nurse who does not respond to the alarm? Testability:
- This system seems critical enough to be tested thoroughly.

Awnser: As stated the server knows which nurses are on duty, so only nurses who are on call for that floor/building will receive an alarm, not a nurse who is on holiday. Also a physical alarm will sound to improve the chance that the closest nurse is noticed. If it is still not noticed, I guess nothing could have and as stated with the measurable this is an acceptable risk.

As for the testing, yes it should be tested properly but it is not the part we have focused on, we have some implications on different scenarios that affect testing and we have reasoned about that but we chose, because of time constrains, to focus on other parts.

E. Thought process

E.1. Mind map high level requirements and functional requirements


Simple conversation
Updates (weather calls etc)
Contact relatives / doctors
Recognize voice
React to emotions
Agenda / Appointments
Video messaging
Entertainment
1. Social interaction
Quality Care Robot
2. Support in everyday live
Assist in making coffee
Answer the door
Vacuum
Waking up
Control thermostat
3. Medical assistance
Provide the right pills at the right time
Insulin shot
Blood analysis
4. Monitoring
5. Reporting
6. Remote controle