FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

This is your Title

Max Mustermann

FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

This is your Title

Dies ist der Titel in Deutsch

Author:	Max Mustermann
Supervisor:	Professor Bernd Bruegge, Ph.D.
Advisor:	Sebastian Peters, M.Sc.
Date:	October 30, 2013

I assure the single handed composition of this master's thesis only supported by declared resources.

Munich, Germany, 2013-10-31 Max Mustermann

Acknowledgments

Thank you, thank you, thank you!

Abstract

One page! What is the content of your thesis? What are the results, what have you achieved?

Common mistake: This is NOT an appetizer to read your whole thesis. The most interesting things should already be in the abstract.

Contents

Outline of the Thesis

CHAPTER 1: INTRODUCTION

The introduction starts with a motivation by describing the current challenges of ...

CHAPTER 2: REQUIREMENTS ELICITATION

Chapter 2 describes the requirements elicitation which includes the scenarios that drive the development of this project. Based on these scenarios, non-functional requirements are proposed and a functional model with use cases and functional requirements is elaborated.

CHAPTER 3: ANALYSIS

Chapter 3 describes an analysis model based on the requirements. The model is created to formalize the objects and information that exist in the domain of ...

CHAPTER 4: SYSTEM DESIGN

Chapter 4 describes the system design model which contains ...

CHAPTER 5: OBJECT DESIGN

Chapter 5 describes ... Afterwards, the transformation of application domain objects to solution domain objects is discussed.

CHAPTER 6: CONCLUSION

Chapter 6 concludes with an overview of the results. Afterwards, the results are reviewed critically and visions for future work are pointed out.

1. Introduction

1.1. Problem Statement

1.1.1. custom subsections here

1.2. Related Work

Here is an example of how it could look like:

To show the latest research in this field, the following section discusses selected literature and related projects.

Tsukada and Yasumura, 2004: UBI-FINGER: A simple gesture input device for mobile and ubiquitous environment, [?]
Tsukada and Yasumura introduce an interface for mobile environments, called *Ubi-Finger*. The concept enables users to sensuously control various fixtures in the real world using finger gestures. [?] The *Ubi-Finger* itself is an input device worn on the fingers. It is connected to a PDA or laptop by a wired serial connection. The target device is defined by infrared sensors attached to each addressable fixture. The actual control commands are transmitted via W-LAN and executed by a server in the background. The gesture recognition is started and stopped using a little button (touch sensor). A target device is then selected by pointing. Afterwards, the device can be controlled with micro-gestures of fingers, like *pushing a switch*, *turning a volume knob*, and so on. [?]
The authors argue that in the communication of human emotions and wills, the non-verbal means were more important than verbal means. Human gestures are considered to be typical examples of non-verbal communication that help people communicate smoothly. They suggest that "human gestures are very useful communication means, and naturally used by everyone". [?] The major benefit of applying gesture input methods to operations of real-world devices is an easy-to-understand mapping of operations with an existing metaphor.
Tsukada and Yasumura's ideas of using pointing and gesturing control mechanisms support the ideas of this thesis. However, there are some disadvantages in the proposed approach. First, the approach requires the installation of additional infrastructure (infrared sensors) on each addressable fixture. Apart from the visual detraction, this solution implies additional costs for the electronic pieces and labor for its installation. Second, the prototype finger is still too big and inconvenient for daily work. The fact that it has to be wired to a PDA might also be distracting for a number of users.

1.3. Terminology

The following section introduces definitions of ambiguous words that are used continuously throughout this thesis.

- **Fixture.** A fixture is an instance of a specific fixture type (see below). An instrumented space consists of a number of fixtures that affect the environmental conditions of the occupant.
- **Fixture type.** A fixture type is the generalization of a fixture. Fixture types in the Intelligent Workplace include:
  - Light
  - Addressable plug load
  - Window blind
  - External louvers
  - Operable window
  - Coolwave
- **Instrumented space.** An instrumented space is an indoor environment with addressable fixtures
- **Smart space.** The word "smart space" is used interchangeably with instrumented space.
- **Mobile device.** A mobile device is the user's smart phone or similar device. The developed prototype of this project works with an iPhone as mobile (control-) device.
- **Occupant.** The occupant is the user of an instrumented space. The overall project purpose is to provide well-being and comfort to the occupant while reducing the consumed energy.
- **User.** The word "user" is used interchangeably with occupant.
- **Facility Manager.** The facility manager is the responsible person for maintenance, care and functionality of a building. This thesis assumes that the facility manager is the administrator of the proposed intuitive controller and cares for the initial set up of the system. Especially in non-commercial smart home environments, it has to be considered that the roles of the building owner, occupant and facility manager are actually taken by the same person.

2. Requirements Elicitation

In requirements elicitation, ... The requirements are gained from the scenarios described below. In general, requirements are divided into functional requirements referring to "interactions between the system and its environment independent" [BD04] and non-functional requirements referring to "user-visible aspects of the system that are not directly related with the functional behavior". [BD04] Pseudo requirements are constraints that were imposed by the client. The proposed intuitive control system is created on the assumption of the requirements elicitation described in the following sections.

2.1. Scenarios

The proposed solution can be applied in different scenarios. For the requirements elicitation, a number of visionary scenarios are created and described below.

Scenario 1: "Example scenario
Mr. Miller enters his office and remains at the door. He sees that the window blinds are lowered and wishes to open them. He opens the HomeGestures app on his iPhone. The app shows that he is now in his office. He presses the big button while pointing to the window and makes a gesture with the iPhone, raising his hand from the bottom of the window to the top. The system opens the window blinds and Mr. Miller proceeds to his desk.

....

Boundary Scenario 4: "Initial Configuration"
Ms. Cooper has moved to a new office and wants to set up HomeGestures for the first time. The administrator has already set up her room with its addressable fixtures on the server using a simple configuration file.
Ms. Cooper opens the HomeGestures app on her iPhone. As her room is yet unknown, the device estimates her position in an adjacent room. She corrects this in the configuration tab by changing the room manually. The system is now taking WiFi measurements (called fingerprints) in the background. Ms Cooper chooses the "Learn fixtures" option and sees all addressable fixtures in her room by the given names of the administrator. She chooses her desk light first, points the iPhone to the light and presses the big button with the "Learn" caption. The app confirms with a sound and Ms. Cooper proceeds analogously with her other fixtures. The office is now configured and can be controlled by any authorized

device.

2.2. Non-functional Requirements

The non-functional requirements (NF) of ... are described in the following subsections below.

2.2.1. NF1: My requirement

Explanation

2.2.2. NF2: My requirement 2

Explanation

2.3. Pseudo Requirements

The following subsections describe the pseudo requirements (PR) of ...

2.3.1. PR1: Platform

...

2.4. Functional Model

2.4.1. Use Cases

A use case is the generalization of all scenarios for a given piece of functionality and helps to find out the requirements of the product. The identified use cases are summarized in the UML use case diagram below (fig. 2.1).

The identified use cases (UC) are refined in the following subsections.

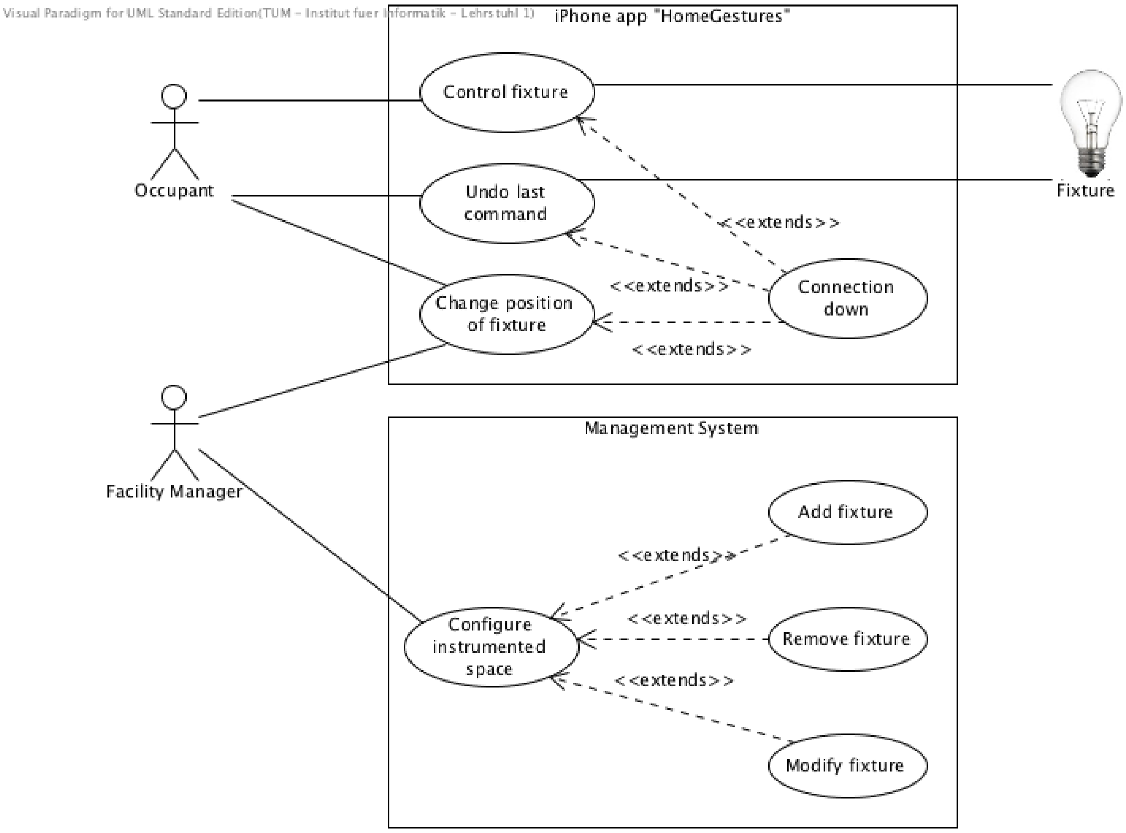

Figure 2.1.: UML use case diagram

UC1: Control fixture

Use case name	Control fixture
Participating actor	Occupant, Fixture
Entry condition	1. The occupant launches the HomeGestures app to control a fixture
Flow of events	2. The app has been launched and has identified the current position 3. The occupant points the smart phone at a fixture, presses the control button and optionally completes a fixture-specific gesture.
Exit condition	4. The state of the fixture has changed to the desired state.
Special requirements	The status of the fixture changes at a maximum of 6 seconds after the user request has been performed. Otherwise, jump into the "Connection down" use case.

...

2.4.2. Functional Requirements

The following functional requirements (FR) have been identified from the use cases above.

FR1: My functionl requirement

Description

....

2.5. Summary

The requirements elicitation chapter describes the overall purpose of ... as it is introduced in the previous chapter. The use cases are based on three main scenarios, that is, the office scenario, the couch at home, the disabled person, and the boundary scenario that describes the initial configuration of the system. The scenarios are then generalized into use cases. From these use cases, functional-, non-functional- and pseudo requirements are gained. The proposed framework for smart home and office environments is built on the assumption of these requirements.
In the next chapter, the information and objects of the application domain are formalized in an analysis model.

3. Analysis

Based on the preceded requirements elicitation, an analysis model is created to formalize the information and objects that exist in the application domain of .... The object-oriented analysis consists of an analysis object model and a dynamic model. [BD04] However, this thesis is not a documentation of the implemented prototype but describes the general concepts and practices in this application domain. Therefore, the formal dynamic model is out of the scope of this thesis. The analysis of the intuitive control system starts with the analysis object model.

3.1. Analysis Object Model

3.2. Summary

Summarize your chapter here

4. System Design

In the system design chapter, the previously discussed analysis model is transformed into a system design model. This model includes a clear description of design goals, a subsystem decomposition and strategies for building the system, such as the hardware/software mapping. [BD04] The following sections describe the system design with strategies and practices of .... Most of the discussed patterns, strategies and practices can be transferred to similar projects in this application domain.

4.1. Design Goals

The intuitive control system design is driven by the following design goals:

1. **Support of different protocols.** Multiple control vendors for the same fixture type are common among smart spaces, such as the Intelligent Workplace. Unfortunately, standards are not used consistently and proprietary protocols are still common practice. Therefore, it is necessary to support different control protocols. For instance, an electric light might be controlled by a proprietary protocol of the vendor, while a window blind is controlled by the common BACnet protocol. It is even possible that two lights are addressed by different protocols. The overall system architecture has to deal with these legacy factors that cause problems.

   ....

4.2. Subsystem Decomposition

4.3. HW/SW Mapping

4.4. Persistent Data Management

4.5. Access Control and Security

4.6. Summary

5. Object Design

In the following chapter we refine three application domain concepts in more detail, that is, ... We also provide a precise view of the corresponding solution domain objects and their interfaces to each other. ...

5.1. Custom sections here

5.2. Custom sections here

5.3. From the Application Domain to the Solution Domain

The previous sections approached issues from the application domain. The following section describes the transition to the solution domain model. In object design, the Broker-concept which has been described earlier in the subsystem decomposition, is now refined in more detail. In addition, the object and subsystem interfaces of the broker are specified, off-the-shelf components are selected, and the object model is restructured to attain the discussed design goals.

5.3.1. Selection of Programming Language and Off-the-Shelf Components

5.3.2. Interface Specification

5.4. Summary

In the object design chapter,

....

. Afterwards, the transformation of the broker's application domain objects to solution domain objects is discussed by specifying object and subsystem interfaces, and selecting off-the-shelf components.

6. Conclusion

6.1. Results

6.2. Case Study at ....?

6.3. Discussion

In the following section, the proposed system is discussed critically and advantages as well as disadvantages are pointed out.

...

6.4. Future Work

...

Appendix

A. This is my appendix chapter

Bibliography

[BD04] Bernd Brügge and Allen H. Dutoit. *Object-oriented software engineering - using UML, patterns and Java (2nd ed.)*. Prentice Hall, 2004.