The UbiLens Approach

Visualisation of and Interaction with Real World Objects through a Mobile Phone's Camera

by
Vina Novitasari Wibowo

Media Informatics
Department of Computer Science
Rheinisch-Westfälische Technische Hochschule
Aachen
Germany

30 April 2010

The UbiLens Approach

Visualisation of and Interaction with Real World Objects through a Mobile Phone's Camera

by
Vina Novitasari Wibowo

A thesis submitted for the degree of
Master of Science in Media Informatics

Supervisors:
Prof. Wolfgang Prinz, PhD.
Prof. Dr. Reinhard Oppermann

Media Informatics
Department of Computer Science
Rheinisch-Westfälische Technische Hochschule
Aachen
Germany

30 April 2010

Layout based on a design by the Media Computing Group,
RWTH Aachen University

http://hci.rwth-aachen.de/karrer_thesistemplate

I hereby declare that I have created this work completely on my own and used no other sources or tools than the ones listed, and that I have marked any citations accordingly.

Cologne, April 2010
Vina Novitasari Wibowo

Contents

Abstract xxi

Acknowledgements xxiii

1 Introduction 1

1.1 Background Motivation 1

1.2 UbiLens at a Glance 3

1.3 UbiLens Usage Scenarios 3

1.3.1 Meeting with Clients 4

1.3.2 Movie Day 4

1.4 Thesis Overview 5

2 Research Assignment 7

2.1 Thesis Claim 7

2.2 Aims and Objectives 7

2.3 Research Questions 8

2.4 Research Methodology 8

2.4.1 User-Centred Design 9

2.4.2 Iterative and Incremental Development Cycle with Prototyping 9

3 Related Work 13

3.1 Augmented Reality 13

3.1.1 Magic Lens 14

3.1.2 WikEye 14

3.1.3 Wikitude World Browser 16

3.1.4 Layar Reality Browser 18

3.2 Real World Objects Interaction 21

3.2.1 kooaba 21

3.2.2 Sweep and Point & Shoot 22

3.2.3 Contextual Bookmarks 23

3.2.4 PERvasive ServiCe Interaction 24

3.3 Object Recognition 28

3.3.1 Marker-based Object Recognition . . . 28

Physical Marker 28

Digital Marker 30

3.3.2 Marker-less Object Recognition 32

Image Recognition 32

Location Recognition 33

4 Knowing the Users 35

4.1 User Profile 35

4.1.1 Who are the Users? 35

4.1.2 What do Users Need? 36

4.2 Preliminary User Questionnaire 36

4.2.1 Questionnaire Settings 38

4.2.2 Participants Profile 38

4.2.3 Questionnaire Result 40

Searching for Smart Objects 40

Visualisation of Smart Objects 40

Selection of Smart Objects 45

Visualisation of Smart Object Functions 47

Information Exchange between Smart Objects 47

Other Requirements 48

4.3 Requirement Analysis 48

4.3.1 Volere Requirement Resources 48

4.3.2 Requirement Analysis Based on Volere 49

5 Early Evaluation with Paper Prototypes 53

5.1 About Paper Prototyping 54

5.2 UbiLens Paper Prototype Evaluation 55

5.2.1 Gathering Materials 55

5.2.2 Evaluation Settings 55

5.2.3 Participants Profile 56

5.2.4 Task Evaluation Result 57

Task 1 - Searching for and selection of object 57

Task 2 - De-selection of smart object . 60

Task 3 - Hidden objects 60

Task 4 - Drag&Drop 61

Task 5 - Create bookmarks 62

Task 6 - Open bookmarks 63

5.2.5 Icon Evaluation Result 63

Application Icons 64

Smart Object Category Icons 64

Hidden Objects Icons 64

Placement of Main Menu Icons 67

5.2.6 Interview with the Left-handed 67

5.2.7 Contribution and Changes to Design . 69

6 From Paper to Reality **71**

6.1 System Architecture 71

6.1.1 Data Transmission 72

6.2 Smart Object 74

6.3 Smart Object Web Service 74

6.4 Object Recognition Component 76

6.5 Object Information Component 78
6.6 UbiLens Mobile Application 79
6.6.1 Android as the Base 79
6.6.2 Visualisation of Real World Objects . 80
6.6.3 Interaction with Real World Objects . 82
Searching for Smart Objects 83
Interaction using Menus 84
Bookmarking 85
Drag&Drop 86
6.7 A Spin-off: Hydragizer 88

7 Time to Evaluate 93
7.1 The Importance of Evaluation 93
7.2 Evaluation Settings 94
7.2.1 Two User Groups 94
7.2.2 Tasks to Do 94
7.2.3 Non-tagged vs. Tagged Environments 95
7.2.4 Think Aloud 96
7.2.5 Measurements Taken 99
7.3 Location and Duration 99
7.4 Participants Profile 99
7.5 The Outcome 100
7.5.1 Task-based Evaluation 100

Task 1 & 6: Searching for Smart Objects 100
Task 2: Checking Out the Services Offered 104
Task 3: Take Away Some Smart Objects 106
Task 4: Opening Bookmarks & Task 6: Drag&Drop 107
7.5.2 User Satisfaction Questionnaire 107
SUMI-based Questionnaire 108
Custom Questionnaire 110
7.6 Problems Encountered 112
7.6.1 Technical Problems 112
7.6.2 Usability Problems 113
7.6.3 Miscellaneous Problems 114
7.7 Evaluation Conclusion 115

8 Summary and View of the Future 117
8.1 Summary and Contributions 117
8.1.1 UbiLens Contributions 119
8.2 View of the Future 120

A Preliminary User Questionnaire Form 125

B Requirements Analysis Based on Volere Card 141

C Paper Prototype Evaluation Form 151

D User Evaluation Tasks **167**

E User Evaluation Form **173**

Bibliography **179**

Index **185**

List of Figures

2.1 Iterative and incremental development cycle . 11

3.1 The evolution of augmented reality 15

3.2 WikEye 16

3.3 Wikitude World Browser views 17

3.4 Wikitude.me web interface 19

3.5 Layar Reality Browser reality views 20

3.6 Layar 3D demo screenshots 20

3.7 Interaction with kooaba 22

3.8 Sweep as optical mouse 23

3.9 Point&shoot technique 24

3.10 A prototype of Contextual Bookmarks application 25

3.11 Perci's augmented posters 26

3.12 Physical mobile interactions 27

3.13 QR Code examples 29

3.14 Semacode example 30

3.15 Visual Code Dialer 31

3.16 Visual Code Profiler 31

4.1 What Users Describe vs. What Users Need . 37

4.2 Age distribution among the questionnaire participants 39

4.3 Occupation distribution among the questionnaire participants 39

4.4 Mobile phone brands distribution among the questionnaire participants 41

4.5 Results for searching smart objects 41

4.6 Found smart objects 43

4.7 Results for visualisation of smart objects . . . 43

4.8 Occluded smart objects 44

4.9 Results for visualisation of occluded smart objects 44

4.10 Out of range smart objects 45

4.11 Results for visualisation of out-of-range smart objects 45

4.12 Results for selection of smart objects 46

4.13 Volere requirement card 50

5.1 UbiLens paper prototype 56

5.2 Paper prototype evaluation sessions 58

5.3 Icon for hidden audio object 67

5.4 Original screen menu icons arrangement . . . 68

5.5 Suggested screen menu icons arrangement . . 68

5.6 Android G1 Phone 69

6.1 UbiLens system architecture 72

6.2 Play video from poster sequence diagram . . 75

6.3 XML object description example 76

6.4 Layers in UbiLens' UI 81

6.5 Camera View layer 81

6.6 UbiLens View layer 82

6.7 Dragging an object 83

6.8 Drop sub-layer 83

6.9 Smart object's menu 85

6.10 Add bookmark by dragging smart object to bookmark drawer 86

6.11 Smart object menu's without bookmark . . . 87

6.12 Compatibility of Drag&Drop objects 89

6.13 XML object description of a movie poster . . 90

6.14 XML object description of a TV 91

7.1 Non-tagged environment 97

7.2 Tagged environment 98

7.3 Age distribution among the evaluation participants 101

7.4 Occupation distribution among the evaluation participants 101

7.5 Participants during evaluation session 102

7.6 A participant is pointing at the QR code . . . 105

7.7 SUMI-based user satisfaction questionnaire result 109

7.8 User satisfaction questionnaire result 111

7.9 Jumping icons due to different reference images114

List of Tables

3.1 Comparison of physical mobile interaction . . 27

4.1 Early user requirements - Functional 51

4.2 Early user requirements - Non Functional . . 52

5.1 Application icons 65

5.2 Smart object category icons 66

7.1 List of non-tagged smart objects 96

7.2 List of tagged smart objects 97

7.3 Participants performance in searching for smart objects 104

Abstract

Ubiquitous computing suggested that computers would disappear and be embedded in everyday life objects so that people do not see them as computers anymore. Digital watches, air conditioners, and mobile phones are among the disappearing computers which can be found today. Further in the future, everyday life objects, such as cups, tables, and kettles, shall also be enhanced with communication, processing, and sensing abilities. With recent technology, these objects are hard to realise. One way to simulate this is by attaching virtual services to everyday life objects. Thus, using an ordinary movie poster, a person is able to buy a cinema ticket, for instance. One might ask how people can locate these objects and consume their services. UbiLens claimed to bridge the interaction between users, real world objects, and their services. It tells the users when objects have services attached and enables users to interact with them. With user centred design and iterative and incremental methodologies as guidelines, users were involved throughout the development process. Studies at the end of the project revealed that UbiLens fulfilled its claim and the participants enjoyed their sessions with UbiLens.

Acknowledgements

First of all, I would like to say my thanks to God. Without His blessings I would not achieve anything in life.

I would like to thank Prof. Wolfgang Prinz PhD and Prof. Dr. Reinhard Oppermann for sparing their time and being my grading professors.

Thanks to René Reiners, my advisor, proof reader, and arguing companion, for the *cues*, ideas, and guidance throughout this thesis project.

Many thanks to Dr. Andreas Zimmermann who has recommended me to work at Fraunhofer and helped me finding my thesis topic.

I would like to express my gratitude to Ferry Pramudianto for helping me going through some technical crisis, to Marco Jahn and Niels Henze for providing me with technical components that built up this thesis project, and also to Marc Jentsch for his ideas, advice, and help.

Of course, I would not even study here in Germany without my father, Santoso Wibowo, and my mother, Marisa Wibowo, who encouraged and financed me to study abroad. Special thanks to both my sisters, Eveline Aprilisa Wibowo and Silvia Aurelia Wibowo who were constantly asking me when I would be finished with my thesis and go home. Somehow I took it as an encouragement to finish this thesis as soon as possible.

I owe my deepest gratitude to my other half, Eddi Siswoyo, who has always been there through thick and thin. He was always cheering me up when I had difficulties with my thesis.

I also like to say my gratitude to all my colleagues and friends for the ideas and mental support given during the development of this project.

Last but not least, I owe big thanks to the individuals who have participated in any of the user studies needed to complete this master thesis. Without their time, ideas, and comments, this thesis project would not be complete.

Chapter 1

Introduction

"This is the beginning of something great."

—Mollie Becker

Everything happens for a reason. This chapter describes the motivation in starting this thesis project. It also presents a short description on what the proposed solution to the current problems is. Two scenarios strengthen the importance of this project. Last but not least, an overview of this thesis is given.

1.1 Background Motivation

The disappearing computers

Fifteen years ago, Mark Weiser coined the term Ubiquitous Computing - the disappearing computers [Weiser, 1995]. He viewed the future to be filled with computers that are embedded into the real world. They fade into the background so that people do not see them as computers anymore. Nowadays, these objects can be found in everyday life. Examples include digital watches, air conditioners, parking assistant in cars, and the infamous mobile phones.

Hyper Objects

Looking at the future, the concept of disappearing computers will be much stronger. Ordinary objects, such as toothbrushes, water kettles, cups, and tables, shall be enhanced with communication, processing, and sensing abilities. Mavrommati and Kameas [2003] called these objects, *Hyper Objects*. With Hyper Objects, it is hoped that they will help people to better fulfil their needs. Consider a hyper cup filled with tea. When the tea level inside the cup has reached a low level. The hyper cup communicates with a hyper kettle and asks the kettle to start boiling water. Thus, when the tea is finished, some hot water is ready to be made into more tea.

Enhancing ordinary objects with virtual services

The question raised here is when these Hyper Objects will finally be available in real life. What can be and have been done at the moment is to enhance these ordinary objects with virtual services [Henze et al., 2008]. Virtual services that allow people to download trailer from a movie poster or to buy a book from a book cover, to name a few. Compared to the Hyper Objects of which sensors and computing power are physically attached to the objects, the attached virtual services are located elsewhere.

Reference to the virtual services

Movie posters often state the URL where people are able to watch the movie trailer or read information regarding the movie on the Internet. This URL serves as a reference to the corresponding virtual service. In this case, the reference is visible. However, this is not always the case. Additionally, in reality, not every real world object has virtual services attached to it. Hence, there should be a way to help people determining whether objects have virtual services. Therefore, people need a tool to help them finding these objects and making use of the services offered.

Mobile phone's popularity

Nowadays, people bring mobile phones wherever they go and manage everyday tasks with them without considering them as computers. This makes mobile phones as the first truly pervasive computing device [Ballagas et al., 2006]. According to the European Information Technology Observatory, the number of mobile phone users would reach 4.4 billion in 2009 [European Information Technology Observatory, 2009]. This prediction was updated by the United Nations that more than 5 billion people were expected to use mobile phones in 2010 [United Nations Radio, 2010]. This popularity increases the demand of having more advanced mobile phones. Many features, such as networking, image recognition, GPS localisation, augmented reality, and many more, were added to answer this call and are widely available on today's mobile phones.

With the popularity and the available technology, mobile phones have great potentials to bridge the interaction between people, real world objects and their virtual services. Mobile phones have been supplied with more and more computing power and characteristics that similar to computers. Therefore, they are able to recognise and communicate with these digitally-enhanced objects.

Mobile phone is a candidate to mediate physical and virtual world

Augmented reality applications have been developed as one of the answers to bridge the interaction between digital and physical worlds [Mobilizy, 2009a][SPRX Mobile, 2009a]. Image recognition applications also try to give users the digital data corresponding to the pictures taken [Kooaba AG, 2009] and save the reference to the digital data for later use [Henze et al., 2008]. Even posters could be used as substitutes to ticket machines [Broll et al., 2007].

Communication with real world efforts

1.2 UbiLens at a Glance

Combining augmented reality, image recognition, and web services, UbiLens allows people to interact with the information or services hidden within physical objects. It is a mobile phone application that mediates the interaction between users and real world objects with services attached to them (i.e smart objects). Keeping in mind that with the idea of disappearing computers, smart objects are intermingled with the other ordinary objects, UbiLens helps users to spot these smart objects. Furthermore, it allows users to consume the services that these objects offer. Complete details on how UbiLens works can be found in section 6—"From Paper to Reality".

UbiLens the mediator

1.3 UbiLens Usage Scenarios

UbiLens can be used anytime anywhere. Two scenarios below illustrate examples on how UbiLens can be used to interact with smart objects in everyday life.

1.3.1 Meeting with Clients

Samantha, 30, is the Vice President of RealEstate Co. Just last month the company installed a new infrastructure which allows physical objects to be tagged with information regarding the services they offer which can be accessed by anyone who installed UbiLens on his/her mobile phone.

UbiLens in presentation

Today, Samantha is going to give a presentation to her client about an upcoming project. Thirty minutes before the presentation starts, she walks into the presentation room. She launches UbiLens which has been installed on her G1 phone. After a few seconds, her phone's camera gets the picture of the room. Then, she pans it across the whole room. While doing so, she sees through the phone's screen that some objects are surrounded by yellow line along the border of the object. She spots that the projector in the room has a yellow border as well. She taps on the screen where the projector is displayed. The screen freezes and displays a set of menu which she can choose. She selects the "Play Presentation" menu. The application asks which presentation file is going to be played. She browses through all her files and selects the presentation file she needs for today. After she confirms her selection, four large buttons appear of the screen: "Previous", "Stop", "Start", and "Next". She taps on "Start" and the projector starts to show her presentation slide and the "Start" button transforms into "Pause" button.

Samantha's client and other guests are arriving. She starts the presentation and controls the playback of the presentation slides by tapping the desired button. After a while, she successfully finishes her presentation and taps on the "Stop" button to end her presentation.

1.3.2 Movie Day

James is a 17 year old teenager who goes to Hope High School. The school has an infrastructure which allows physical objects to be tagged with information regarding the services they offer which can be accessed by anyone who installed UbiLens to his/her mobile phone.

Each month the school gathers all students in the main hall to watch a documentary movie together. Just two weeks before the movie day, a poster displaying four movies is placed on the cafeteria's wall. Each student can vote for a movie to be played. Of course, this poster is one of the objects which is tagged with movie trailers.

UbiLens in multimedia

James is standing in front of the poster. He launches UbiLens. After a few seconds, the phone's screen displays the picture of the cafeteria. He points his phone camera at one of the movie pictures on the poster. He sees that an icon is overlaying the movie picture. He touches the icon from his mobile phone's screen and slides it to the corner of the screen. This copies the movie icon and displayed it at the corner of the screen. He looks around and sees that there is a TV set nearby. Pointing the camera at the TV set, he sees that it has another overlaying icon. He drags the movie icon at the corner of the screen to the TV. The TV set then plays the movie trailer that he has chosen.

1.4 Thesis Overview

Chapter 1—"Introduction"

Chapter 1—"Introduction" opens up this thesis with the background and short introduction of the system. Scenarios were also added to strengthen the importance of using the system, as well as to shape the idea of UbiLens interaction.

Chapter 2—"Research Assignment"

Chapter 2—"Research Assignment" describes the purpose of this thesis project along with the questions that had to be answered near the end of the project. It also presents the research methodologies which have been used as guidelines to control the flow of this project.

Chapter 3—"Related Work"

This project is not brand new but rather combination of existing technologies. Chapter 3—"Related Work" summarises several technologies that made up this thesis project.

Chapter 4—"Knowing the Users"

Along with the chosen research methodologies, users were involved from the beginning of this thesis project. Chapter 4—"Knowing the Users" presents the first user involvement with the preliminary user questionnaire. This questionnaire was used to get early ideas on what the system should do or look like.

Chapter 5—"Early Evaluation with Paper Prototypes"

The preliminary user questionnaire was followed by paper prototyping sessions. These sessions, described in chapter 5—"Early Evaluation with Paper Prototypes", were necessary to get feedback from users in interacting with the system. Even though the prototype was in the form of paper, a significant result was achieved.

Chapter 6—"From Paper to Reality"

After the results of paper prototype evaluation were analyse, implementation phase began with still keeping users' needs in mind. Chapter 6—"From Paper to Reality" explains the details of the UbiLens implementation including the components architecture and communication between them.

Chapter 7: Time to Evaluate

Before ending this thesis project, user evaluation needs to be done in order to validate the feasibility of the prototype in terms of its usability. Evaluation sessions and the result will be presented in chapter 7—"Time to Evaluate".

Chapter 8: Summary and View to the Future

Chapter 8—"Summary and View of the Future" summarises the main points of this thesis work. It also includes the contribution that this thesis has made to the society. Several possible future works is presented as well in order to address certain problems found during the development of this project and also to keep this research going on.

Chapter 2

Research Assignment

"There is nothing like looking, if you want to find something. You certainly usually find something, if you look, but it is not always quite the something you were after."

—J.R.R. Tolkien

Every project needs a description to guide where it should go in order not to get lost. This chapter describes the scope of this thesis project.

2.1 Thesis Claim

UbiLens as mediator

This thesis claims that UbiLens is an alternative approach to mediate interactions between users and physical objects in consuming virtual services offered by the objects. These interactions shall be done via a mobile camera phone.

2.2 Aims and Objectives

The project goal

The aim of this thesis project is to promote interactions in smart environments using a mobile camera phone. Following the aim of this thesis project, several objectives were set.

These objectives include but are not limited to the following ones.

- To *investigate* the possible interactions in smart environments using mobile camera phones.
- To *build* prototypes for supporting interactions in smart environments using mobile camera phones.
- To *evaluate* the prototypes along with the possible interactions with users.

2.3 Research Questions

Questions to be answered

As mentioned in 2.1—“Thesis Claim”, the main challenge of this thesis project is to facilitate the interaction between user and physical object through a mobile camera phone. Upon the completion of this thesis project, the following questions shall be answered.

1. *How can smart objects be recognised?*
2. *Which visual methods are effective in giving cues that an object has services attached?*
3. *What visual cues are effective to indicate that the objects are occluded by other objects?*
4. *How can selection ambiguity be solved when two or more smart objects are closed together?*
5. *How do users interact with smart objects via a mobile phone?*
6. *What kind of interactions are possible in smart environments?*

2.4 Research Methodology

User-centred design and iterative-incremental development techniques combined with prototyping were chosen as research methodologies to guide the development process of this thesis.

2.4.1 User-Centred Design

Norman's user-centred design

Focusing on the interaction between users and physical objects, this thesis project adopted the *User-Centred Design* methodology. Below is what Norman [2002] said about User-Centred Design.

> *"User-centred design is a philosophy based on the needs and interests of the user, with an emphasis on making products usable and understandable."*

Continuous user involvement

Users were involved throughout the entire development process of this thesis. This involvement started with the preliminary user questionnaire which was published on the internet to gather early user requirements on this thesis topic. Following this questionnaire, paper prototype evaluation was conducted in order to evaluate the early user interface and interaction design of this application. Further user involvements were done throughout the development process in the form of software prototype evaluation sessions and usability questionnaires.

2.4.2 Iterative and Incremental Development Cycle with Prototyping

Iteration to improve usability

In user-centred design, users are involved throughout the software development life cycle. Starting at the beginning of the development cycle, developers should ask users what they want in a system. However, it is nearly impossible to finalise the user requirements for the whole system at the beginning of the development process [Dix et al., 2004]. Therefore, the process should be separated into several parts followed by user testings for each part. After each user testing, several refinements are necessary to improve the usability of the system. At least three versions of refinement are needed to improve the usability of the system [Nielsen, 1993].

Division of the project into parts and cycles

Iterative and incremental development cycle with prototyping was selected to guide the development process of this thesis. The idea of this method is to separate the whole system into smaller parts in which each part has its own iterative cycle. Prototypes are introduced to the users before the system is actually implemented. Figure 2.1 depicts the iterative and incremental development cycle.

Project flow

This thesis project has been divided into three cycles. The project started off with a preliminary user questionnaire that was published online in order to gather user requirements mostly regarding the user interface and interaction of UbiLens. This completes the first cycle. Further details of this questionnaire could be found in section 4.2—"Preliminary User Questionnaire". The second cycle dealt with paper prototype and its user evaluation (cf. 5.2—"UbiLens Paper Prototype Evaluation"). The third cycle included the development of the software prototype (cf. 6—"From Paper to Reality")along with its user evaluation which was held at Fraunhofer Sankt Augustin (cf. 7—"Time to Evaluate").

Hydragizer

A spin-off of UbiLens was also implemented in between. It was developed as a part of Hydra middleware[1] demonstrator called *Hydragizer* that was exhibited at CeBIT 2010 in Hannover. Further details on Hydragizer can be found in section 6.7—"A Spin-off: Hydragizer".

[1]http://www.hydramiddleware.eu/news.php

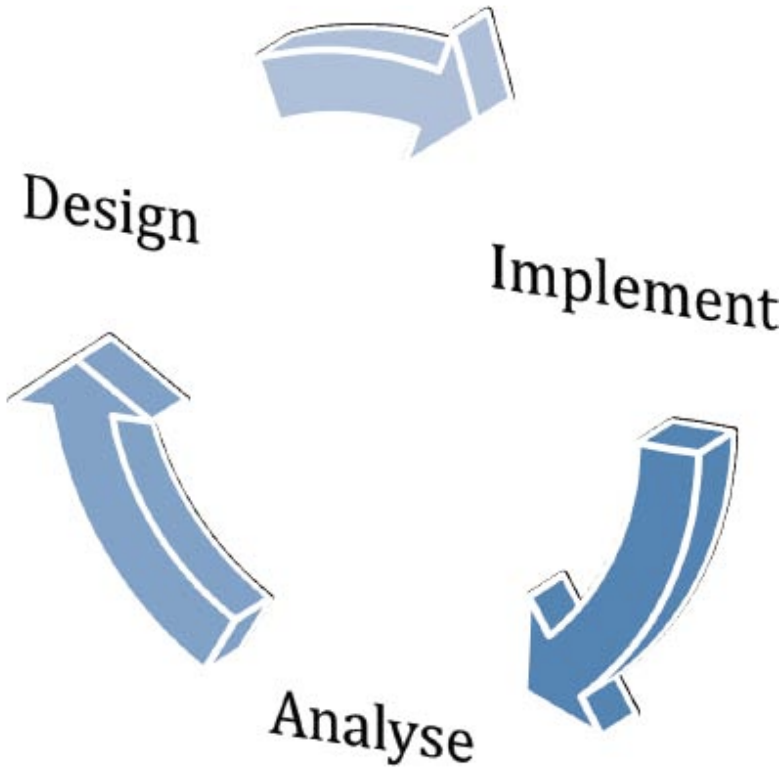


Figure 2.1: Iterative and incremental development cycle. Each incremental cycle has its own iterative cycle.

Chapter 3

Related Work

"Work grows out of other work, and there are very few eureka moments."

—Anish Kapoor

The idea to bridge the virtual and the physical world is not new. Up until now, there have been research projects going on how to merge these two worlds. UbiLens introduces an alternative way to interact with real world objects with the help of mediating devices (i.e. mobile phones). Rather than a new approach, it is a combination of three areas: augmented reality 3.1, real world object interaction 3.2, and object recognition 3.3. This chapter describes related research projects done in these three fields.

3.1 Augmented Reality

Augmented reality

Augmented reality (AR) tries to bridge virtual and physical world by enhancing the physical world with virtual objects in order to improve user's perception and interaction with the real world [Azuma, 1997]. In augmented reality, users see both physical and virtual objects co-existed in the same space, in which these virtual objects add values or information to the physical ones. AR has been used in many areas; from helping surgeon performing a liver surgery [Paloc et al., 2004], assisting in object assembly [Tang et al., 2003], to playing games [Herbst et al., 2008][Oda et al., 2008].

History of AR

Back in the early years of AR development, users were required to bring a backpack with a portable PC inside and wear a head-mounted display (HMD), which overlays virtual objects on top of real world view. Since this setting was uncomfortable for the users, AR developers replaced it using ultra mobile PCs (UMPCs). When Personal Digital Assistants (PDAs) and later on smart phones hit the market, they saw this opportunity as a way to make AR available to more customers. Nevertheless, with the limited processing power of smart phones, careful choice of algorithms and optimised code needs to be considered. Figure 3.1 depicts the evolution of AR. [Wagner and Schmalstieg, 2009]

3.1.1 Magic Lens

Magic Lens

Since smart phones become more popular and more powerful these days, many developers are competing to deliver mobile AR applications to the market. Mobile AR transforms mobile phones into a so-called *magic lens* through which users can see beyond the reality. This type of interface was actually introduced in 1993 by Bier et al. [1993]. Back then, the magic lens approach is used to reveal hidden information, enhance data of interest, and suppress unnecessary information on any platform (i.e. PC, notebook, PDA, etc,). Answering the call for mobile AR, this term was adapted by Rohs and Oulasvirta [2008] to be an augmented reality interface that consists of a mobile device with built-in camera acting as a see-through tool.

3.1.2 WikEye

WikEye

Based on the adapted magic lens approach, WikEye was developed to improve the navigation of information of handheld devices and to overcome the display size limitations [Rohs and Oulasvirta, 2008]. WikEye is an augmented reality application which combines the advantages of both paper and digital maps [Hecht et al., 2007].

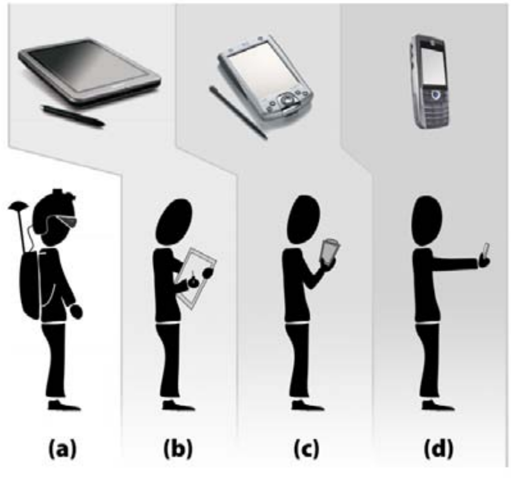


Figure 3.1: The evolution of AR: (a) Backpack with HMD, (b) UMPC , (c) PDA, (d) Smart Phone [Wagner and Schmalstieg, 2009]

Paper vs. digital map

Paper (static) maps have high resolution, large scale, and zero power consumption; yet, the printed information cannot be changed without reprinting the whole map. Meanwhile, digital (dynamic) maps have low resolution, small scale, and consume power; yet, they could display up-to-date and personal information on the map. Combining the positive features of both maps, WikEye overlays dynamic information on top of paper maps, which are captured by the mobile phone's camera.

Algorithm

WikEye computes the position of the camera view by continuously analysing the video stream recorded by the mobile phone's camera. Regularly spaced black dots are placed on the maps and serve as reference points (cf. Figure 3.2). The information displayed is taken from Wikipedia, which has been processed offline to extract the geo-referenced and temporal content.

WikEye interaction techniques

Dedicated to tourism, WikEye accepts three interaction techniques. *Sweeping* allows users to explore the information hidden behind certain points on the paper map. When they reach an interesting point, they can *point&shoot* to view the rest of the Wikipedia article. Time-related article can be chosen by *rotating* the mobile phone until a certain

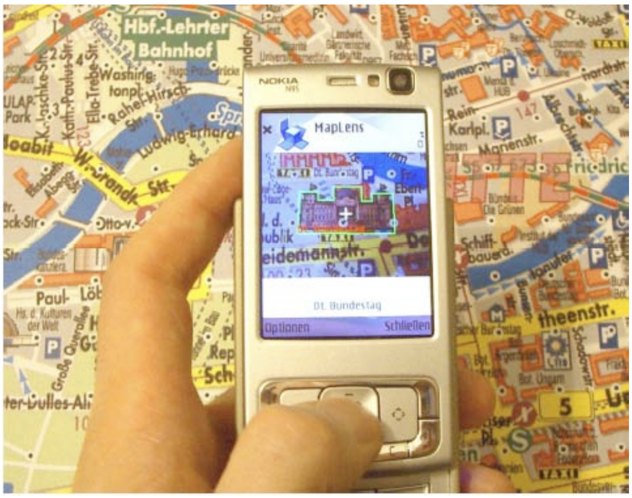


Figure 3.2: WikEye

time period desired is depicted on the phone's screen.

No further interaction

Align with its goal, which is to provide interaction with Wikipedia content on a mobile device as an overlay of physical maps, WikEye does not provide further interactions with the point of interests.

3.1.3 Wikitude World Browser

Wikitude World Browser

Wikitude World Browser is a location-based mobile AR [Mobilizy, 2009a]. Developed by Austrian company called Mobilizy, Wikitude was intended to help users to plan trips or find out landmarks around them [Mobilizy, 2009b].

Information sources

Wikitude provides users with information regarding their current location. The information is taken from three different sources: articles from Wikipedia[1], reviews from Qype[2], and crowd-generated information from Wikitude.me[3] [Parr, 2009].

Available views

The information can be viewed in three different ways: list

[1] http://www.wikipedia.org/
[2] http://www.qype.co.uk
[3] http://www.wikitude.me

Figure 3.3: Wikitude World Browser views: (a) list view, (b) map view, (c)(d) AR cam view

view, map view, and AR cam view (cf. Figure 3.3). With the list view, the information is displayed on a list. This lacks of the information about the direction of the landmarks and users need to browse through a long list before finding what they really need. The map view uses Google Maps with marks on the point-of-interests. The heart of Wikitude lies on the AR cam view. By turning on the camera, the users see both real world live image and digital information through the camera screen.

Sensors used

In order to give users information as accurate as possible, Wikitude relies heavily on the use Global Positioning System (GPS) module and the built-in compass within the mobile phone. The use of GPS drains battery power and does not work indoor. Although not as accurate as GPS, a cell-based positioning system is used as an alternative because it works indoor. When the cell-based positioning system does not return the desired results, users can enter their current address instead. The built-in compass is a must because Wikitude only returns information to the user based on the orientation of the mobile phone's camera.

Wikitude pros and cons

Other than displaying information and directing users to the source website of the information, Wikitude does not provide further interaction with the landmarks. Nevertheless, Wikitude gives an alternative way to browse the web faster by returning information based on the location of the users which is especially great for tourists. Wikitude also encourages people to geo-tag any location in the world by providing Wikitude.me [Mobilizy, 2009c]. Wikitude.me, as one of the information sources of Wikitude, serves as a cost-free yet effective way to gather more information regarding any location in the world (cf. Figure 3.4).

3.1.4 Layar Reality Browser

Layar features

Similar to Wikitude (3.1.3), Layar Reality Browser brings real time location aware digital information on top of reality captured by mobile phone's camera [SPRX Mobile, 2009a]. Even though, as Wikitude, Layar focusses on the enhanced AR view, it still provides user with less interactive list and map views. In the AR view, rather than giving the users with information directly overlaying the points of interest, Layar has an information box at the bottom of the screen. In this way, the information does not occlude other point of interests (cf. Figure 3.5).

Layar = layers

Layar comes from the word "layer(s)". As the name implies, the information shown in Layar is divided into layers. Each layer has a certain category, such as a house-for-sale layer, a popular-bars layer, a tourist-information layer, or a reality-game layer (cf. Figure 3.5). In November 2009 there were 239 layers available in Layar [4]; and this number kept increasing. Users can choose which layer(s) they want to be searched for and displayed on the screen. Since SPRX Mobile, the developer of Layar, provides API for creating their own layers, these layers are sponsored by third party

[4] Two months before there were only 87 layers.

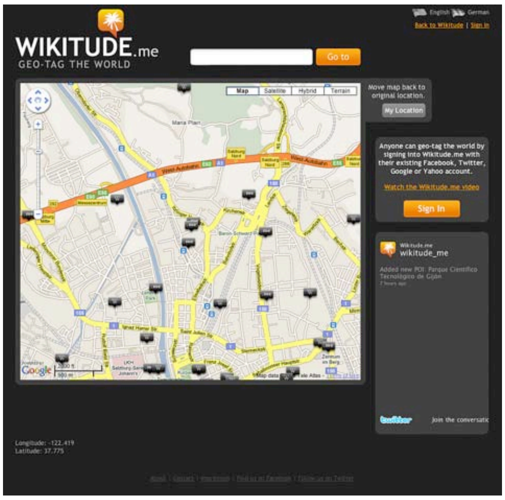


Figure 3.4: Wikitude.me web interface

developers.

Layar sensors

In order to provide with the desired information, Layar uses GPS sensors and compass to determine the location and orientation of the phone. As mentioned in the previous section (3.1.3), the use of GPS drains battery power and it does not work indoor. However, Layar also provides the option to use the less-accurate cell-based positioning. Furthermore, it needs a camera to display the current video live-stream.

Layar in 3D

At the end of November, Layar planned to launch Layar 3D[SPRX Mobile, 2009b]. With Layar 3D, users can see 3D objects and hear sounds about a particular point-of-interests. Let alone seeing, users can interact with these 3D objects. Each 3D object will be associated with ”open link” or ”play music”. These features require the use of OpenGL to draw the 3D objects and accelerometer to detect the viewing angle, as additional sensors to GPS and compass used in the original Layar. Figure 3.6 depicts a couple of demo screenshots of Layar 3D.

Interaction with real world?

Nevertheless, it is still questionable whether the interaction with 3D objects means an interaction with real world objects. One can assume that 3D object should represent real world object. However, looking at the demo screenshots,

Figure 3.5: Layar Reality Browser reality views: (a) houses for sale, (b) bars, (c) museums, (d) reality games)

Figure 3.6: Layar 3D demo screenshots: (a) 3D Pacman, (b) 3D building tagging, (c) 3D fly-by airplane [SPRX Mobile, 2009b]

figure 3.6(a), this is not the case, as Pacman and the ghosts cannot refer to real world objects.

Layar expectations

Layar is available on Android mobile phones and iPhone 3GS. It provides an alternative way to browse through surrounding information of which amount increases rapidly as many third party developers are competing to compose their own layers. With the Layar 3D, developers hope that it will provide richer interaction.

3.2 Real World Objects Interaction

Real world objects

The previous section (3.1) focusses more on the discovery of landmarks and their related information. Nevertheless, real world objects are not limited to landmarks. Ordinary things, such as books, movie posters, public displays, and public transportation, are categorised as real world objects. These objects are linked to virtual services which enable users to purchase books, watch movie trailers, or purchase transportation tickets. Several interactions below have been proposed in order to interact with real world objects via mobile phones.

3.2.1 kooaba

Point-snap-find

kooaba is a mobile application running on Android phones and iPhones that recognises certain types of real world objects and allows users to further interact with them [Kooaba AG, 2009]. What users need to do in order to use their service is to *point* their phone's camera to a certain object, take a picture of it (*snap*), and wait until the result is returned by the system (*find*) (cf. Figure 3.7). Not only information is returned, but also step-by-step interactions on how users can watch movie trailers, purchase songs, DVDs, CDs, or buy movie tickets.

Smart visual

The digital image sent from the users' device is converted into a so-called Smart Visual using kooaba's image recognition services. A smart visual is a "smarter" version of the digital image. After being converted, the smart visual can be explored, shared, and organised.

Further development

Till Quack, the CTO of Kooaba AG, mentioned that kooaba was developed to provide end-users with mobile visual search. In November 2009, kooaba was able to recognised book cover, CD covers, DVD, movie posters, game covers, and a limited selection of magazines. He foresaw that in the near future, kooaba would extend the range of recognisable print medias and be able to recognise landmark buildings and scenery [Amazon Web Services, 2009].

Figure 3.7: Interaction with kooaba: (a) *point* and *snap* a DVD cover, (b) *find* possible matches, (c) choose possible interactions, (d) purchase the DVD from ebay

Try-and-wait

Kooaba maintains its own matching database to match users' image with the corresponding information and possible interaction. The limitation of the database results in limited data available. Users need to try-and-wait before finding out whether kooaba has information regarding their current search query.

3.2.2 Sweep and Point & Shoot

Interact with public displays

Ballagas et al. [2005] proposed two techniques in which people can interact with large public displays. Public display, which is categorised as real world object, is most of the time unavailable for direct-touch interaction due to expensive software and vandalism. Ballagas et al. [2005] saw the opportunity to use mobile phone as an input device based on the fact that people are more comfortable using their own devices. These two techniques are called *sweep* and *point&shoot*.

Sweep

Sweep turns the phone's camera into optical mouse (cf. Figure 3.8). Using optical flow image processing, the algorithm calculates where the current position of the cursor. With this algorithm, users do not need to point the phone's camera at the public display. They can wave the phone in the

Figure 3.8: Sweep as optical mouse [Ballagas et al., 2005]

air and point anywhere. This supports 3-axis interaction. For better experience, users need to stand at some distance from the public display so that the display seems considerably smaller. Otherwise, the users need to stretch their arms to go from one corner to another.

Point&shoot

The *point&shoot* technique needs visual markers to set up the absolute coordinate system on the display surface. It used to determine which object is selected. Figure 3.9 depicts the three steps in this interaction technique. The use of Visual Codes could hinder visualisation of the information. For future use, infrared codes are proposed because they are recognisable by the camera but invisible to human eyes.

Other objects are not supported

This project contributes two new interaction techniques for interacting with the real world. However, besides the interaction with large public displays, interaction with other kinds of real world objects is not supported.

3.2.3 Contextual Bookmarks

Contextual Bookmarks

Contextual Bookmarks consist of a snapshot of a physical object taken with a mobile device and meta-information about the content related to the corresponding physical ob-

Figure 3.9: Point&shoot technique [Ballagas et al., 2005]

ject [Henze et al., 2007]. In other words, it represents a physical object in a digital form. It enables users to capture interesting information of a particular physical object with a mobile camera phone in a so-called bookmark. Users can create, use, and share these bookmarks [Henze et al., 2008].

How Contextual Bookmarks work

A picture of the physical object is taken using mobile phone's camera. This picture, along with GPS and time information, is then sent to a recognition server [Henze et al., 2007]. In the server, this snapshot is mapped to a digital service. Users receive related digital items and services [Henze et al., 2008]. Figure 3.10 illustrates a prototype of Contextual Bookmarks application.

Try-and-error

Similar with kooaba (cf. 3.2.1), Contextual Bookmarks require users to try-and-error. If after several try-and-error sessions, no information is returned, users could refuse to use contextual bookmarks anymore. Therefore, a hint whether an object has its digital form needs to be shared with the users.

3.2.4 PERvasive ServiCe Interaction

Perci ServiCe Interaction (Perci)

PERvasive investigates and develops new techniques of interaction with the Internet of Things [Perci]. It provides a framework to access and interact with web services by interacting with the corresponding real world objects via mobile phones[Broll et al., 2007]. Several realised examples include buying movie tickets from a movie poster and buying train tickets from a public transportation map (cf. Figure 3.11) by using one of the *physical mobile interactions* [Rukzio et al.,

Figure 3.10: A prototype of Contextual Bookmarks application

2006].

Physical mobile interaction

Physical mobile interaction (PMI) is a type of interaction in which the user interacts with a mobile device and the mobile device interacts with real world smart objects [Rukzio et al., 2006]. Acting as a mediator between the users and the smart objects, the mobile device is also comparable with a universal remote controller.

PMI techniques

PMIs enable the use of mobile services attached to the smart objects. There are four different physical mobile interactions: touching, pointing, scanning, and user-mediated interaction. Each has its own advantages and disadvantages. Figure 3.12 depicts the touching, pointing, and user-mediated interaction. Factors, such as the application context, the object's location, and the distance between users and the objects, influence which technique will be chosen by users. [Rukzio et al., 2007]

Touching

Touching requires users to get close to the smart object and bring the mobile device into contact with it (cf. Figure 3.12(a)). It obviously needs more physical effort as it cannot be done when the object is further away. Because of this, according to the evaluation done by Perci, touching is the least favourite compared to other techniques. However,

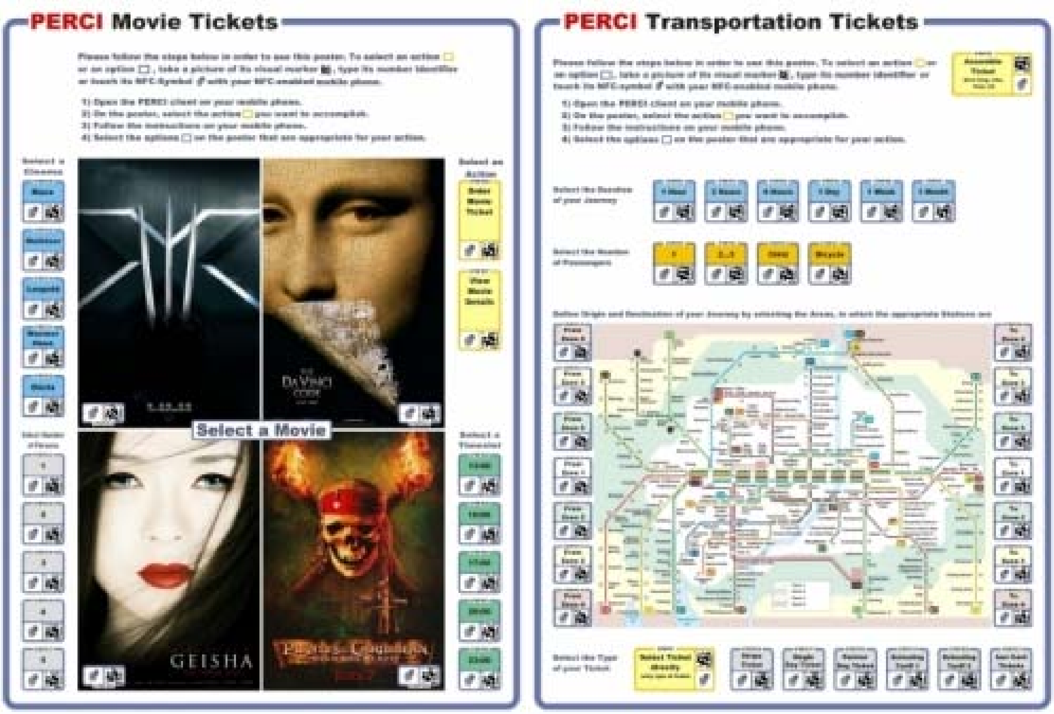


Figure 3.11: Perci's augmented posters [Perci]

touching gives more accuracy for small or close together objects. [Rukzio et al., 2006]

Pointing

On the other hand, *pointing* is the most favourite technique because it is intuitive (comparable with remote controller) and do not need a lot of physical effort since it can be done from distance. Nevertheless, a direct line of sight between the mobile device and the smart object need to be maintained and ambiguity can occur for small or close together objects. *Pointing* is done by aiming the mobile device at the smart object, for example by taking picture of the visual marker of the object (cf. Figure 3.12(b)) [Rukzio et al., 2006].

Try-and-wait

Touching and pointing requires users to touch or point and wait for feedback whether the smart object is recognised before interacting with it. It is possible that the smart object is not recognised at all. Hence, similar to kooaba 3.2.1 and contextual bookmarks 3.2.3, this try-and-wait mechanism can be a burden to the users.

Figure 3.12: Physical mobile interactions: (a) touching, (b) taking picture (pointing), (c) user-mediated interaction [Perci]

Scanning

Scanning returns a list of available nearby smart objects using wireless mechanism. It overcomes the physical distance and the direct line of sight problem [Rukzio et al., 2006]. After being evaluated, the indirectness of scanning makes this interaction avoided by the users [Rukzio et al., 2007], although is still preferred than pointing [Rukzio et al., 2006]. This indirectness lead to difficulty in mapping the list item with the objects in the real world [Rukzio et al., 2007].

User-mediated interaction

User-mediated interaction is probably the most common and used interaction nowadays. In order to interact with a certain smart object, users need to input either the identifier of the object or a certain URL to the mobile device (cf. Figure 3.12(c)). Inputting numbers (identifiers) do not need much effort. This changes when users need to type a lengthy URL. [Rukzio et al., 2007]

Table 3.1 compares the features of four different pmi.

Factor	**Touching**	**Pointing**	**Scanning**	**User-mediated**
Proximity	direct contact	direct line of sight	nearby	does not matter
Accuracy	accurate	less accurate	all objects are listed	very accurate
Example	NFC [a]	light sensor, pictures	Bluetooth	form

Table 3.1: Comparison of physical mobile interaction

[a] Near-Field Communication

3.3 Object Recognition

Object recognition

Before any further interaction, objects need to be recognised. Developers have used several object recognition techniques. Although, how the object recognition is performed is not the concern of UbiLens, a careful decision must be taken to choose which object recognition method would likely enhance the interaction between users and real world objects.

3.3.1 Marker-based Object Recognition

Marker-based recognition

In order to recognise a certain object, marker-based object recognition needs a certain kind of marker which usually placed nearby the object. These markers are divided into two types: physical and digital. Physical markers are usually printed on the object itself and visible to people. Digital markers are located on or nearby the object and may not be visible.

Physical Marker

Visible markers

Physical markers are visible markers printed on the corresponding objects. These markers are placed on a place where people could see them. They should be big enough to be recognisable, yet small so that they do not obstruct or occlude the corresponding objects. A camera, with corresponding recognition application, is required to recognise this type of markers. Several physical markers have been developed with differences in encoding methods.

Introduction to barcode

Barcode is a symbol in the form of light and dark bars [Smith and Offodile, 2002]. These bars are able to store numbers or ASCII characters to identify a certain object. Since early 1970s, barcode has been used in the industry to ensure the accuracy of and speed up the process of product identification.

Linear and 2D barcode

Two types of barcode are available on the market today: *1D* or *linear barcode* and *2D barcode*. The main difference

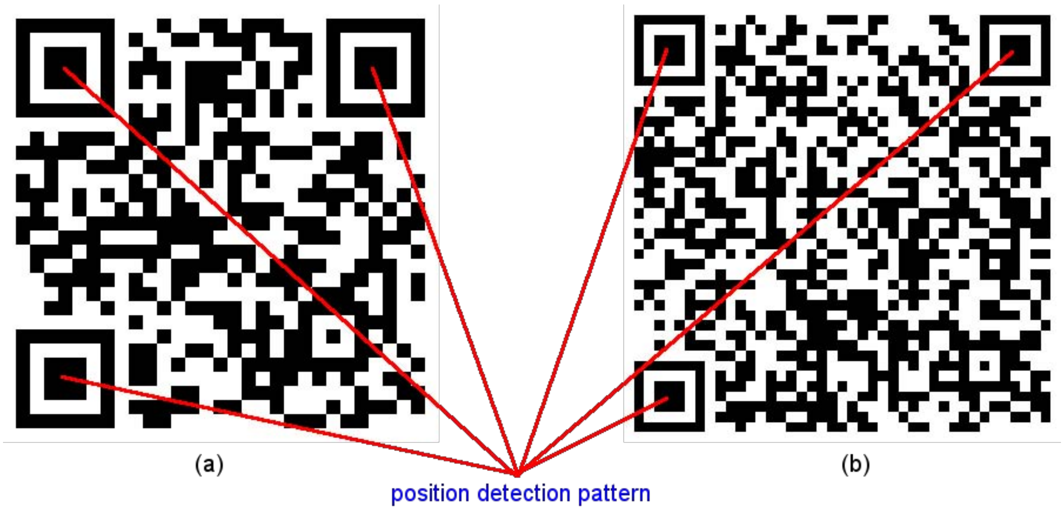


Figure 3.13: QR code examples: (a) URL, (b) Plain Text[a]

[a]QR codes were generated from http://qrcode.kaywa.com/

between the two is the amount of data they can store. Linear barcodes are found on food packagings, book covers, or clothes tags. This type of barcode normally contains product code for simple identification. When more information is to be stored, such as the URL of the product, 2D barcode is chosen. makebarcode.com [2010] compares several standards of both linear and 2D barcodes.

QR code

QR code, a kind of 2D barcode, was first introduced in 1994 by Denso Wave [Denso Wave]. It was developed to improve the limitations of linear barcode. It has more information, more character types, and can be printed in small size to be recognisable. Thanks to the position detection pattern (cf. Figure 3.13), QR code is also rotation resistant; meaning that the code can be rotated and still be recognised. A single QR code stores up to 7,089 characters. QR codes can be generated from either plain text, URL, SMS, or phone number [Kaywa]. They have been used in manufacturing, logistics, and sales. Figure 3.13 illustrates examples of QR code.

Semacode

Similar with QR code, Semacode is a form of 2D barcode. Instead of using lines like in original barcodes, Semacodes use dots representing a website address which can be accessed from mobile phone. It provides the ability to do one

Figure 3.14: URL Semacode example[a]

[a]Semacode was generated from http://semacode.com/tag

click access to websites without typing the lengthy URLs [Semacode Corporation]. Semacode example can be found in figure 3.14.

Visual Code

Concerned with the limited computation power of mobile phone, Rohs and Gfeller [2004] developed Visual Codes which are lightweight for mobile phone because they avoid floating point operation if possible. Along with the Visual Codes, Visual Code Recognizer was built to recognise these codes. Compared to the recognisers of QR Code 3.3.1 and Semacode 3.3.1, which can only recognise one code at a time, Visual Code Recognizer is capable of multiple recognition. Examples using Visual Code include application to dial phone number (cf. Figure 3.15) and set profile on mobile phone (cf. Figure 3.16). Perci 3.2.4 also use this code for *pointing* technique.

Digital Marker

RFID/NFC

Radio-Frequency IDentification (RFID) or especially *Near Field Communication* (NFC) has been used to digitally tag physical objects. Both use radio frequency to connect the

Figure 3.15: Visual Code used in dialing phone number [Gfeller and Rohs, 2006]

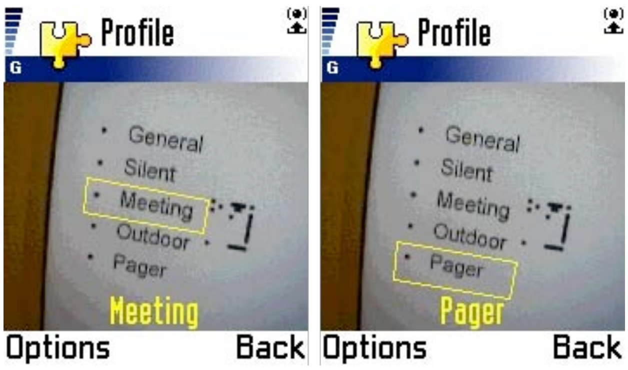


Figure 3.16: Visual Code used in setting mobile phone's profile [Gfeller and Rohs, 2006]

tag and the reader wirelessly. With NFC suits mobile phones better, it promotes promotes short-range, contact-less identification, data-exchange and interaction [Broll et al., 2009].

NFC tag

NFC reader

NFC tag is attached to the corresponding device. This tag contains passive information that is used to identify the object itself. An NFC reader is actively reading the tags to gather the data stored within. This reader can be attached to or built into a mobile phone, for example Nokia 6131 NFC. In order to read the tags, the reader must be located

3 - 5 cm from the tag. Perci 3.2.4 uses NFC tag and reader for *touching* technique. It requires a passive visual cue to the users so that they aware of the location of the tag in order to interact with it. This cue is mostly printed on the object, which is similar to physical marker 3.3.1. This is why physical markers are preferred since no additional device, except camera which is available on almost every mobile phones, is required [Roberts, 2009].

Bluetooth

Bluetooth also offers another way to tag physical object. Similar to NFC 3.3.1, Bluetooth uses short-range radio frequency to connect between Bluetooth-enabled devices [Bluetooth SIG, Inc, 2009]. Using this connection voice and data can be transmitted [MobileInfo.com, 2001]. Until December 2009, many mobile phones were capable of Bluetooth connections and the price of separate Bluetooth devices were getting cheaper. This makes Bluetooth a better candidate to physical object recognition than NFC. Nevertheless, since Bluetooth device can detect other devices in 10 - 100 metres range, it cannot be used for a precise interaction. Users need to choose a particular object from a list of discovered Bluetooth devices in order to interact with it.

3.3.2 Marker-less Object Recognition

Marker-less object recognition

Marker-less object recognition recognises objects without the need of physical or digital marker. Instead, it uses the feature of the object itself.

Image Recognition

How image recognition works

Compared to the physical markers mentioned above 3.3.1, in this case, the image recognition algorithm is used to recognise the object itself and not the marker placed on the object. An image of the object is processed and matched to the corresponding object from the database. The process can be done on the mobile device or on a dedicated server. The later option is better since more data can be inserted into the database without altering the client application. Example applications are kooaba 3.2.1 and Contextual Bookmarks

3.2.3.

Image recognition is prone to problems, such as occlusion – when other objects partly conceal the image –, cropping – when the user only take a part of the picture –, and clutter – when the image contains the object and background. In order to avoid these, kooaba 3.2.1 uses an algorithm that - rather than matching a whole image - determines regions of an image that are then used to match against the image sent by the user [Kooaba AG, 2009].

Problems occured

Location Recognition

Location-based recognition is another object recognition technique that is growing in popularity. This recognition technique is mainly used to recognise objects with fixed GPS coordinates. Landmarks, buildings, restaurants, and houses are among the recognisable objects. In order to use this recognition technique, a GPS receiver is required. Because of its popularity, many mobile phone manufacturers are starting to build the receiver into the mobile phones. Applications using location based recognition, such as Wikitude 3.1.3 and Layar 3.1.4, are becoming a hit.

GPS

Location-based, especially GPS-based, recognition does not work accurately indoor. An alternative, such as cell-based, does work indoor although not as accurate as GPS-based (cf. 3.1.3). Additional compasses were built into mobile phones to enhance the accuracy of the location-based recognition, especially for augmented reality applications such as Wikitude 3.1.3 and Layar 3.1.4.

GPS enhancement

Chapter 4

Knowing the Users

"Know thy user, and you are not thy user."

—Arnie Lund

UbiLens tried to involve users throughout every step of the development process. This chapter, in line with the development cycle mentioned in 2.4.2—"Iterative and Incremental Development Cycle with Prototyping", presents the first cycle of UbiLens development. In this cycle, users were involved in order to gather the early requirements UbiLens should have.

4.1 User Profile

Who the users are and what they need

Knowing what users need is the first step towards a successful interaction design [Dix et al., 2004]. Before knowing what they need, developers need to first define who they are. Are they old or young? Do they have experiences in such system? What do they do for a living? These also hold true for a new interaction technique, such as UbiLens.

4.1.1 Who are the Users?

UbiLens' users

UbiLens, as a mobile phone application that bridges virtual and physical world, imagines its users to be people who own and actively use smart phones. It is not dedicated to a certain group of users. However, since the concept of having UbiLens is quite new, it might be more suitable for young people only because they tend to be more open to try new applications or interaction techniques compared to the old ones. Nevertheless, it welcomes all users from any age who are willing to maximise the use of mobile phones, instead of just using them for calling and texting.

4.1.2 What do Users Need?

Avoid misunderstanding

After defining who the users of the system are, knowing what they need with the system is the next step before starting to develop the system. Several methods such as interview and observation about their life are used to capture what users really need since most of the time describing what a system should do lead to a misunderstanding which then lead to an unused application [Dix et al., 2004]. Figure 4.1 depicts a failure of a system that might happen because of misunderstanding what users need in the early stage of development.

Requirements gathering with online questionnaire

UbiLens' users consist of individuals who have different backgrounds. Interviewing or observing each of these individuals or groups of individuals personally is not an option to gather user requirements. Therefore, an online questionnaire was chosen. It could not only grab different groups of participants but also more participants. This increases the quality and quantity of the requirements gathering.

4.2 Preliminary User Questionnaire

The goal

The primary goal of the preliminary user questionnaire is to gather user requirements. Based on this goal several objectives are defined as below.

- To cover more variety of potential users in terms of background, age, gender, occupation, and so on.

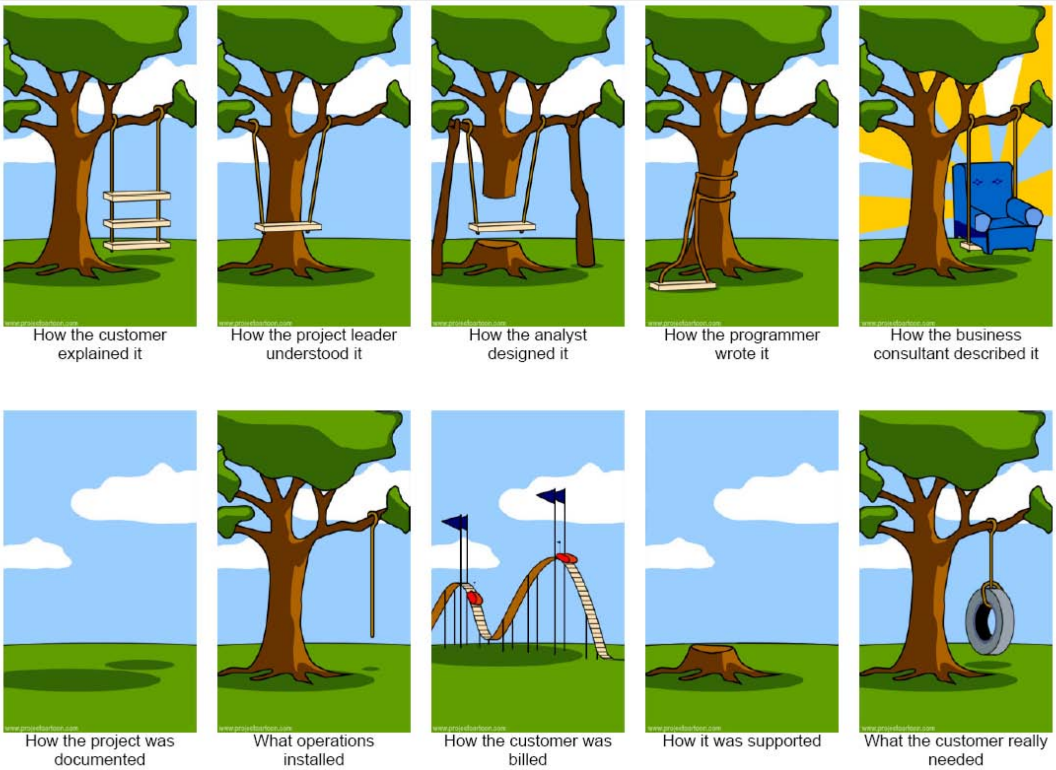


Figure 4.1: What Users Describe vs What Users Need [a]

[a] Taken from http://www.jacobsen.no/anders/blog/

- To cover more potential users.
- To find out how the participants would interact with UbiLens.
- To find out which user interface participants most familiar with in terms of operating systems and mobile applications.

Is the system needed?

In order to avoid a system being useless, especially for systems which are built based on other systems, the first question to ask the users is whether they actually need it. UbiLens is based on a collection of unrelated systems (cf. Chapter 3—"Related Work") combined to realise a new concept. Therefore, the question whether it is needed was omitted because the idea here is to introduce this new concept. Instead, the questionnaire focused on the appearance or user interface of UbiLens.

4.2.1 Questionnaire Settings

The content

The questionnaire begins with a short description of what UbiLens is (at that time, the application was called XApp). A set of possible techniques in interacting with UbiLens were presented to the participants in the form of figures to make everything clear. The participants are required to choose which technique is the most suitable given a certain situation. An additional option is also included in order to encourage the participants to suggest other possible techniques. A complete listing of the questionnaire can be found in Appendix A—"Preliminary User Questionnaire Form".

The publication

The questionnaire was posted online[1] within two different periods, from 15 - 22 June 2009 and 23 - 30 June 2009. The questionnaire in the second period was the revision of the first version. This was done due to the fact that there were several feedbacks given to improve the questionnaire. The questionnaire was designed and published using an online survey tool called QuestionPro[2] with Student Research Sponsorship[3].

4.2.2 Participants Profile

Age and gender

60 people started this questionnaire. From this number, only 40 participants fully completed the questionnaire. This created difficulty in filtering the result of the questionnaire. From these 40 participants, 23 participants were male and 17 were female. Their age ranged between 18 and 40 years old (cf. Figure 4.2) with an average of 11.5 years of experience in using mobile phone.

Occupation

Most of the participants are students and computer scientists (cf. Figure 4.3), in which 16 participants do not have Computer Science backgrounds. From 24 participants with Computer Science backgrounds, 14 have developed mobile applications before.

[1] http://vina.thesis.pre.questionpro.com
[2] http://www.questionpro.com
[3] With Student Research Sponsorship a student can use the full version of the survey tool for free for a duration of six months.

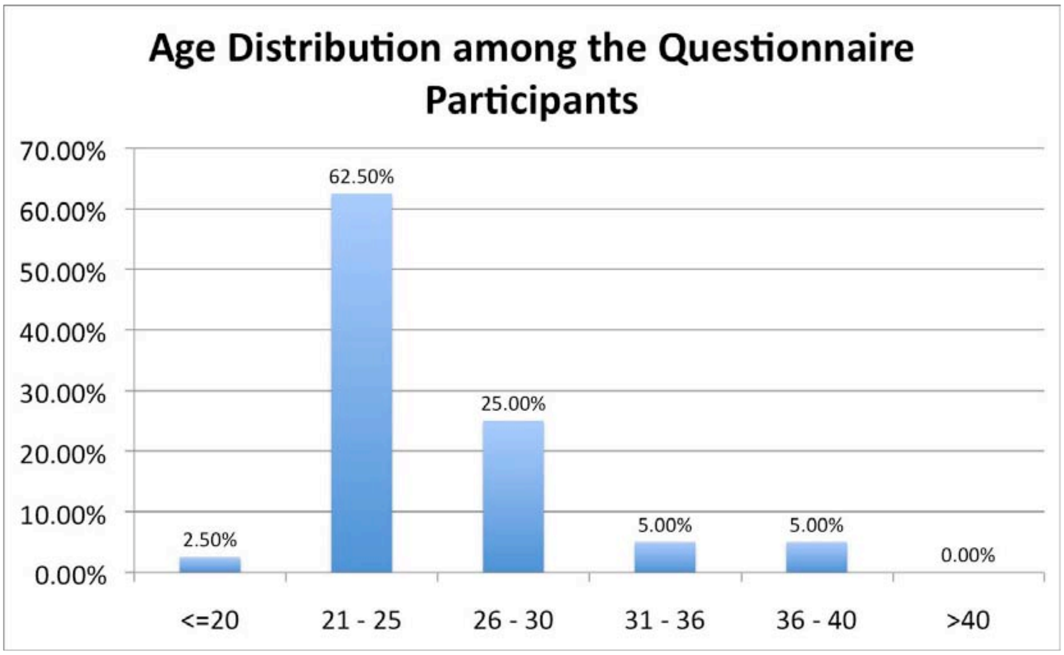


Figure 4.2: Age distribution among the questionnaire participants

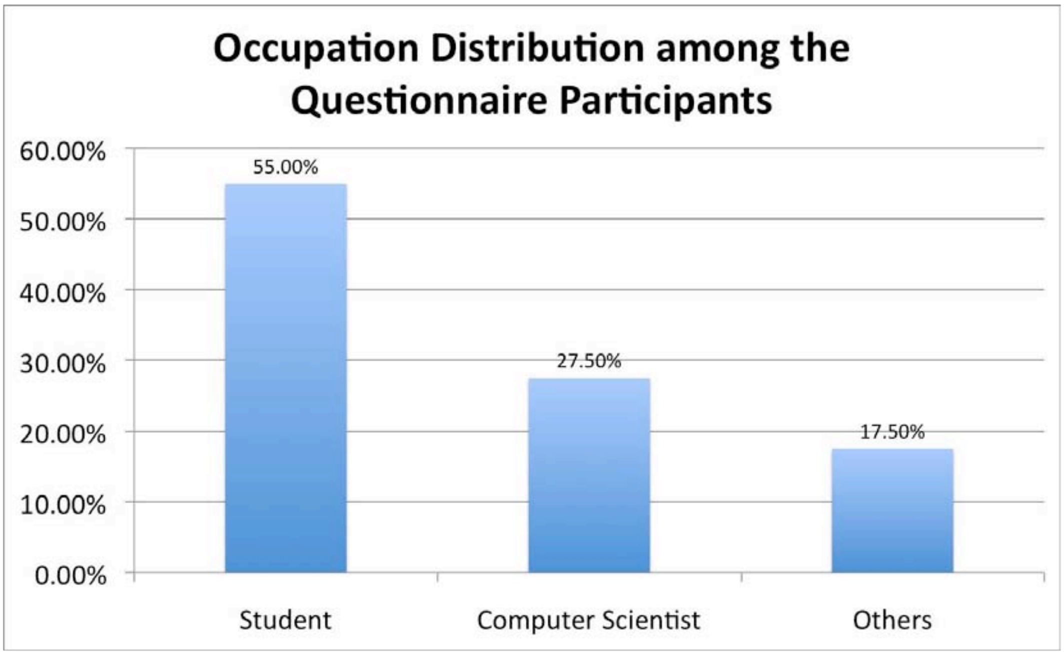


Figure 4.3: Occupation distribution among the questionnaire participants

The most used mobile phones and applications

Nokia and Sony Ericsson have proven to be the most used mobile phone brands, with 33 and 26 participants respectively (cf. Figure 4.4). SMS placed first as the most used application, followed by call application. Advanced applications, such as camera, Internet, and GPS/Map, were considered less used. Since the user interfaces and interactions among different operating systems and applications are different, these properties need to be taken into consideration in developing UbiLens.

4.2.3 Questionnaire Result

Categorical result

The result of the questionnaire is presented in categorical form depending on the situation in which the interactions are used.

Searching for Smart Objects

How to search for smart objects

Smart objects need to be searched and found before any further interaction with them is possible. The participants were given three possibilities on how to do this. These options include panning the camera (mobile phone's camera) through the whole room, getting closer to the object and focussing the camera to the object for a couple of seconds, and taking pictures of the corresponding object.

Panning the camera is chosen

Among the three possibilities, more than half of the participants chose to search for smart objects by panning the camera (cf. Figure 4.5). As alternatives, two participants suggested having a list of nearby smart objects; while another participant mentioned about using RFID (reader).

Visualisation of Smart Objects

Border vs icon hightlights

Whenever smart objects are found during the searching process, UbiLens needs to alert the user. Among the six options given, three frame smart objects with coloured borders while the other three highlight them with icons. Both coloured borders and icons differ in the uniqueness of the border or

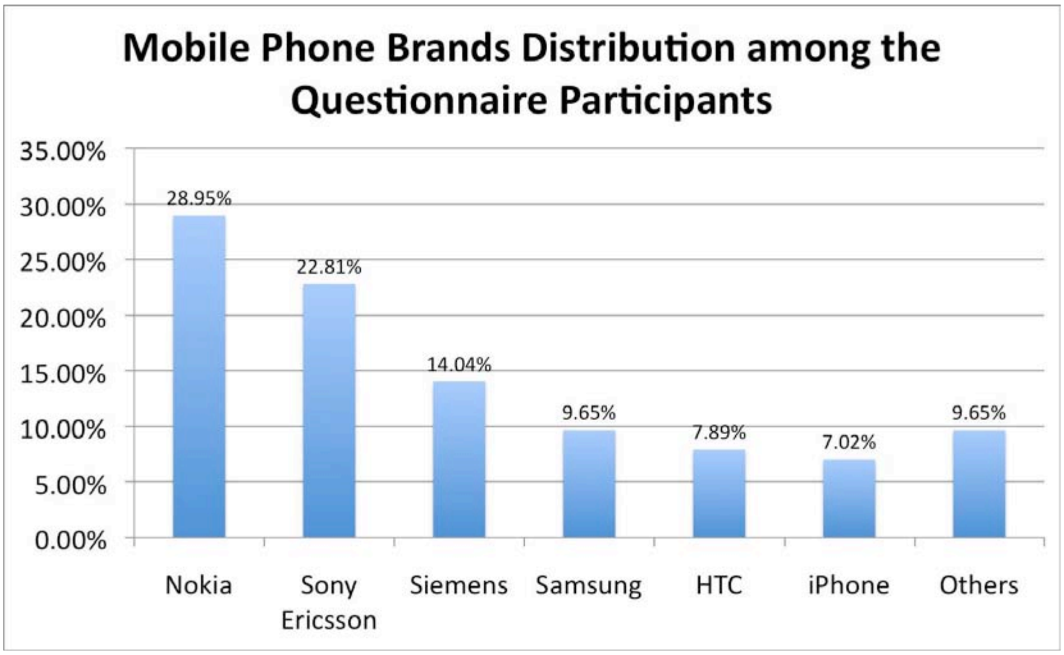


Figure 4.4: Mobile phone brands distribution among the questionnaire participants

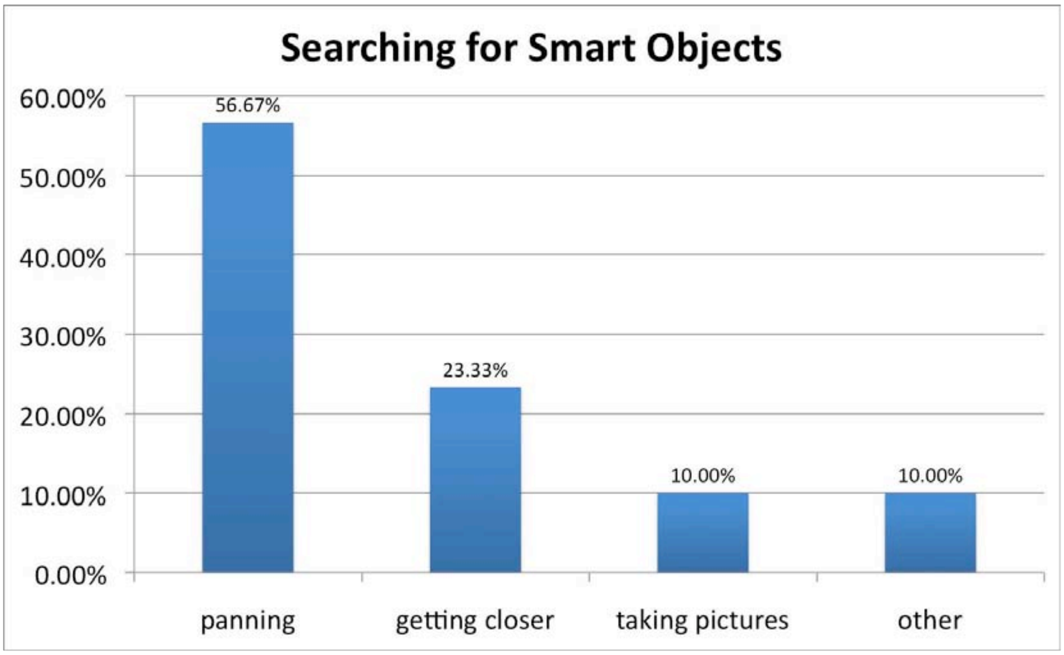


Figure 4.5: Results for searching smart objects

icon. These include coloured borders or icons which are common for all objects, unique for each object, or the same for smart objects with the same categories.

Coloured border was preferred

Based on nearly 70% of the participants, smart objects should be indicated by surrounding them with coloured border (cf. Figure 4.6(a)). Nevertheless, in the first version of this questionnaire, there was no further question asking whether the participants would like to have same colour for all objects, unique colour for each object, or categorical colour for objects with similar functions. Upon second installment of this questionnaire, the options regarding the border were improved to address this issue. However, which border was the "winner" cannot be decided since the participants taking the second version of this questionnaire could not represent the whole participants. These options would be later investigated during paper prototyping. The rest of the participants preferred using icons to indicate smart objects; with 22.64% went for common icons (cf. Figure 4.6(b)). Additionally, one participant mentioned that the borders or icons should not be displayed on the screen; instead, displaying them using the internal beamer of the phone directly on the object itself. Figure 4.7 summarises the choice of visualisation techniques.

Occluded objects

Sometimes other smart objects, other objects, or people occlude a smart object and cannot be see through the phone's screen. Again the options were divided into two categories: coloured borders (with the same or different colour for visible and occluded objects) and icons (with different coloured icons, transparent icons, or different icons).

Dashed border was preferred

43.40% of the participants chose to indicate the hidden smart object using dashed border (cf. Figure 4.8(a)). For participants who chose to have icons instead of borders, different colour icons were preferred to indicate hidden smart objects (cf. Figure 4.8(b)). Another approach mentioned by one participant was to have the appropriate icon (either common, unique, or categorical) along with an additional symbol, such as a small arrow, to indicate that there is a hidden smart object. The participant, who mentioned about having a list of nearby smart objects, mentioned that by having a list, this occlusion problem should not be a problem at all. A summary of the alternatives can be seen in figure

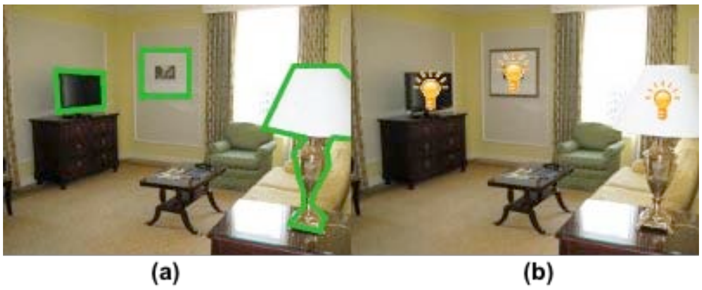


Figure 4.6: Found smart objects are (a) framed with coloured borders or (b) highlighted with icons

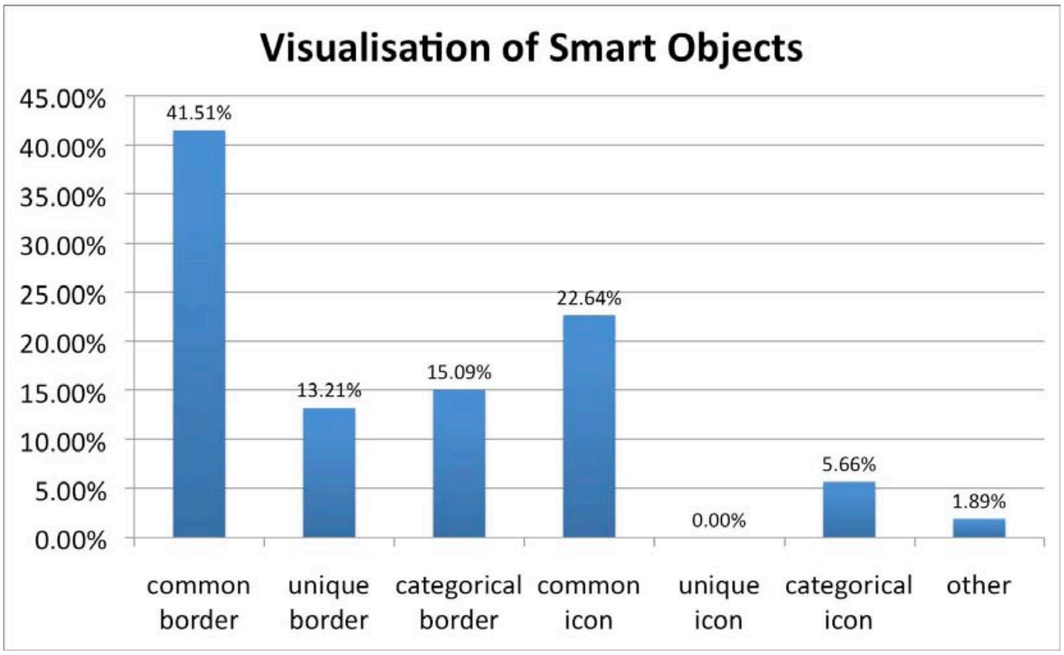


Figure 4.7: Results for visualisation of smart objects

4.9.

Off-screen objects

The view range of the phone's camera is very limited. A mechanism showing that there are other objects nearby the current view range could help users to find the objects faster. The participants were asked whether they need to know that there are other objects outside the phone's camera range. For participants who want this feature, options whether to use arrows (cf. Figure 4.10) or halos were given.

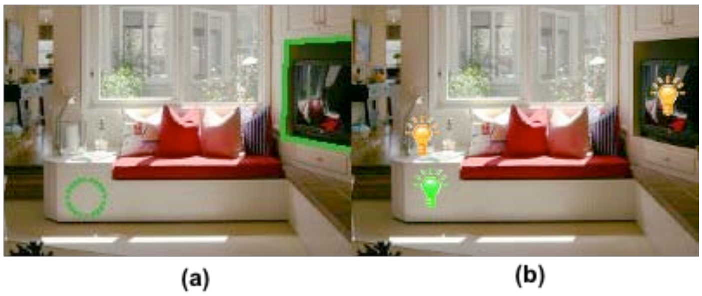


Figure 4.8: Occluded smart objects are indicated using (a) dashed border or (b) different-colour icon

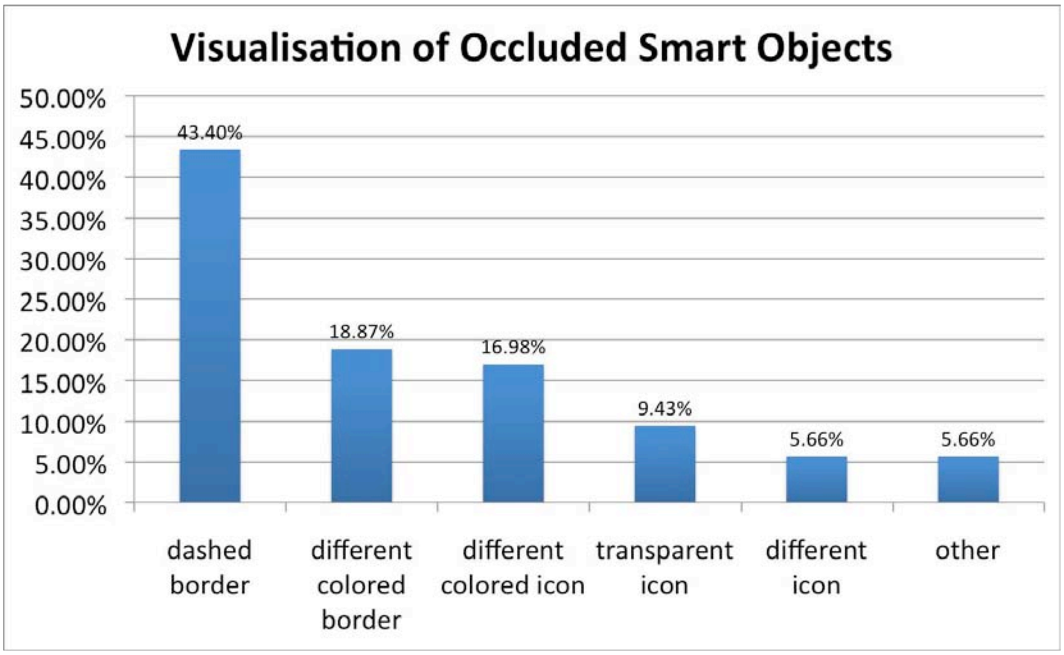


Figure 4.9: Results for visualisation of occluded smart objects

arrow to point out objects

Nearly half of the participants preferred to have an arrow to indicate that there is a smart object located out of the camera's view range (cf. Figure 4.11). A quarter wanted to have halos instead, while the other quarter said they did not need such mechanism. The participant, who mentioned mentioned about having a list to show all smart objects in the area, again pointed out the importance of a list in addressing this problem.

Figure 4.10: Arrows indicated that there are other smart objects outside the mobile phone's camera's view range

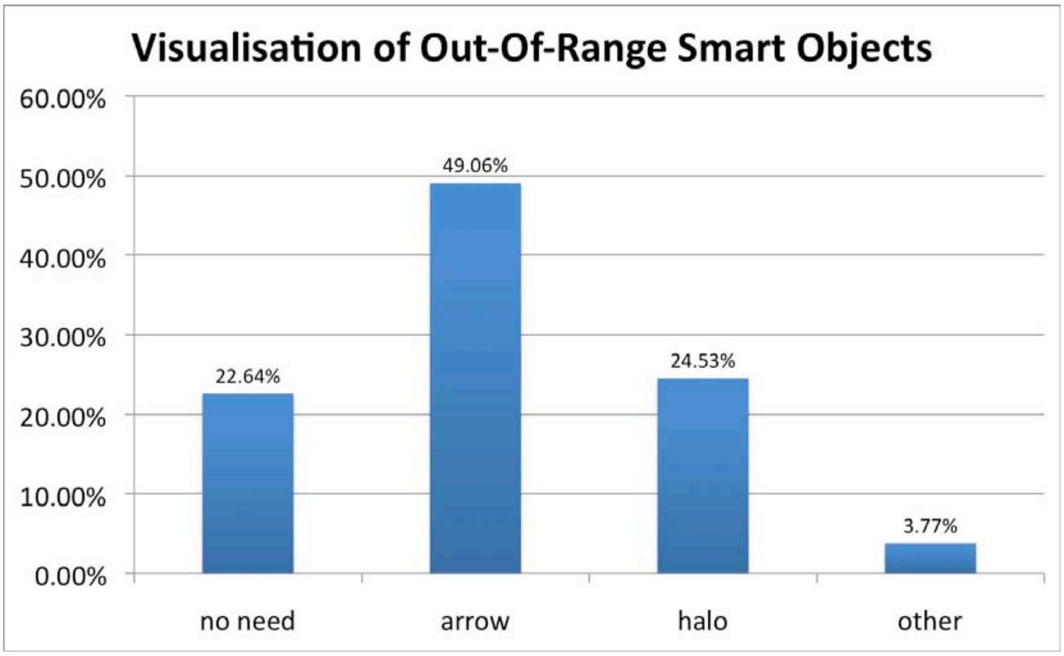


Figure 4.11: Results for visualisation of out-of-range smart objects

Selection of Smart Objects

Touching vs taking pictures

Once the smart objects are found, users need to select one of them to start interacting with it. The participants were given two options by which they could select a smart object. First is by touching the smart object via the touch screen and second is by taking picture of the smart object. With the emergence of touch-sensitive mobile phones, 83.67% of the participants preferred to directly touch the object dis-

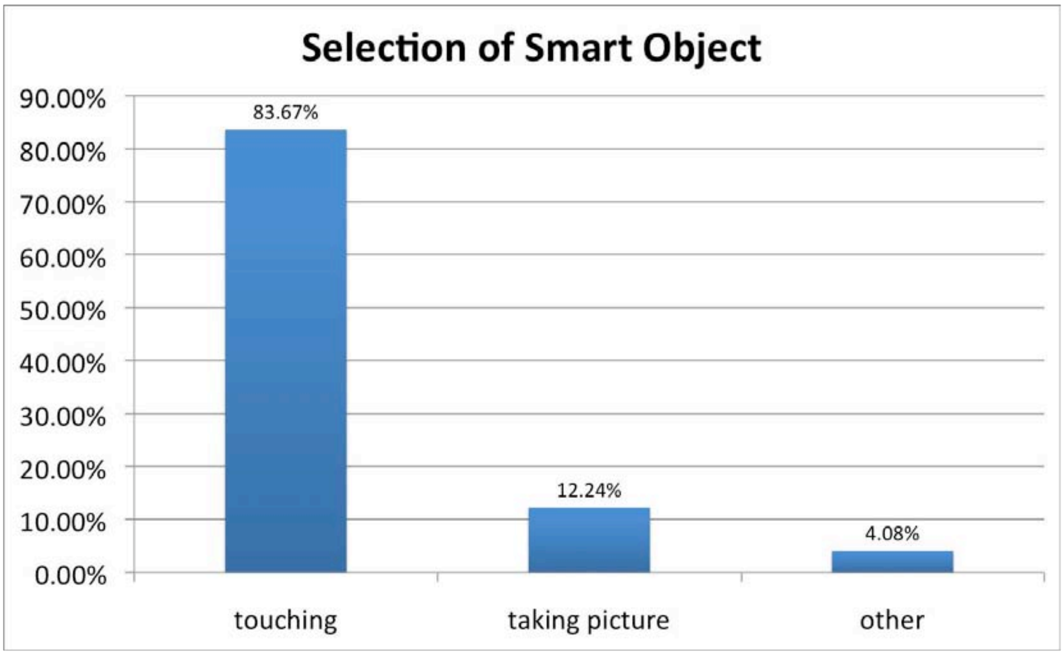


Figure 4.12: Results for selection of smart objects

played on the screen (cf. Figure 4.12).

After selection

The participants were also asked whether after selecting a smart object, a list of functions were expected. In addition, the issue whether the camera image should freezes was raised. 85% of the participants were satisfied with the idea of having a list of the device functions or menus and freezing camera image underneath. The others came up with several suggestions listed below.

- To have the menu without freezing the camera image.
- To have the camera image in grayscale except the one chosen.
- To have the chosen electronic devices automatically turned on.

De-selection of a chosen objet

At some point, it is possible that the smart object chosen is not the smart object wanted. Here several possible gestures, such as dust-sweeping and X-figure gesture, are given. However, 70% of the participants still preferred to use the standard back or cancel buttons. Other suggestions include, touching the smart object again, quickly tilting the phone, double clicking, shaking the phone, and doing circle gesture on the screen.

Visualisation of Smart Object Functions

Menu layout

The menu of the smart object consists of its function, description, and other related information. 37.50% of the participants preferred to decide whether they would like to have this menu to be displayed using buttons or list. One participant mentioned that it does not matter which layout is chosen as long as the layout suits the interaction concept of the phone.

Tilting to navigate

Newer mobile phones are equipped with accelerometers which could open a new interaction metaphor. In this project, one possibility is that by tilting the phone, the users could navigate through the menus. However, based on the questionnaire, no one chose to use this tilting gesture; instead 70% of the participants chose to tap the menu item directly on the screen.

Information Exchange between Smart Objects

Sending information

In a smart environment, it is possible to send information, such as files and hyperlinks, from one smart object to another. Two questions were asked to get some ideas from the participants how this transferring mechanism is done.

Together seen objects

The first question was whether touching the first object on the screen with finger, dragging the finger to the second object, and releasing the finger was a possible solution for this mechanism when the two objects are seen on the screen. 85% said that it is possible. However, two participants said that it is not convenient to do it while holding the phone still and also might be embarrassing to do it in public places. One participant mentioned that a list of information was preferred, supposed that the smart object has more than one information to share.

Far away objects

The second question dealt when the two objects are not seen together on the screen. More than half of the participants preferred to transfer information by touching the first object on the screen with finger, moving the phone to the direction of the second object so that the finger touches the second object on the screen, then releasing the finger. However, one participant pointed out that continuously touching the object while moving the phone was difficult. He then suggested that it could be done by touching the first object for three seconds to *lock* it and releasing the finger, moving the phone, and touching the second object. Another suggestion was to store the information in a temporary storage then transfer. This is in line with the idea of Contextual Bookmarks by Henze et al. [2007].

Other Requirements

Basis for requirement analysis

Apart from answering the questions given, the participants were given the opportunity to mention requirements that they think UbiLens should have. These requirements and the questions asked during the preliminary user questionnaire were used as the base of the requirement analysis process.

4.3 Requirement Analysis

Help in decision process

The result of the questionnaire and the requirements mentioned by the questionnaire participants were gathered and analysed. This analysis helped to decide whether a particular requirement needed to be realised during the development process.

4.3.1 Volere Requirement Resources

Volere requirement card

Volere Requirements Resources [Atlantic Systems Guild Ltd., 2009] composed a template to help system designers to analyse the requirements gathered from (prospective) users. This template is often printed on a card for the ease of distribution between system designers and developers in a project. Figure 4.13 illustrates the Volere requirement card along with descriptions for each field [Robertson and Robertson, 1999]. Each card represents one requirement.

4.3.2 Requirement Analysis Based on Volere

requirement analysis summary

Table 4.1 and 4.2 summarise several user requirements that need to be taken care of when developing UbiLens. A complete listing of the user requirements analysis based on Volere card can be found in Appendix B—"Requirements Analysis Based on Volere Card"

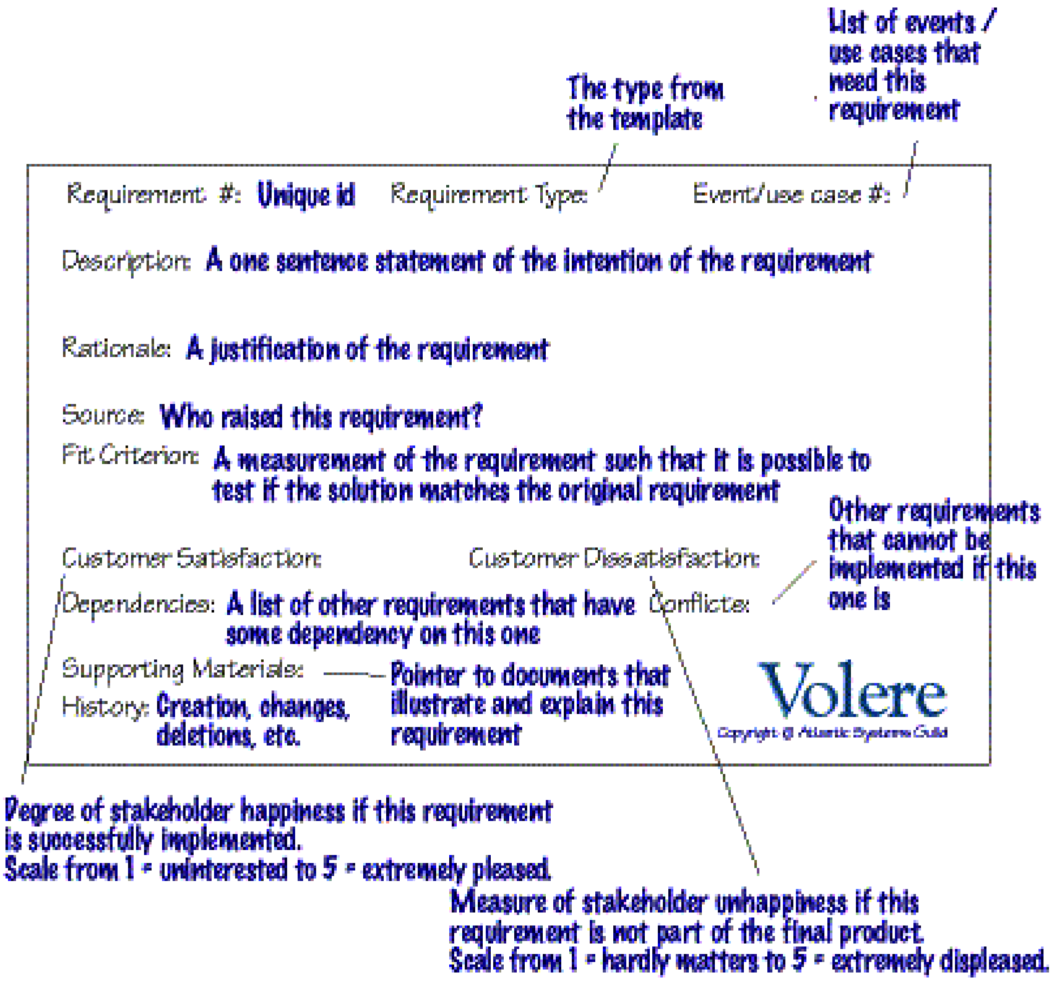


Figure 4.13: Volere requirement card

Number	Description	Category	CSR[a]	CDR[b]
#001	The system shall help users to find service-enabled objects.	F[c]	5	5
#003	The system shall give hints to users whenever it finds visible service-enabled objects.	F	5	5
#004	The system shall give hints to users whenever it finds hidden service-enabled objects.	F	3	2
#005	The system shall give hints to users whenever it finds service-enabled objects outside the users' view range.	F	3	1
#006	The system shall provide mechanisms for interaction between users and service-enabled objects.	F	5	5
#007	The system shall facilitate information exchanges between service-enabled objects.	F	4	4
#008	The system shall help the user in deciding which service-enabled object to use in a certain situation.	F	3	1
#010	The system shall provide mechanism to consume services from service-enabled objects anytime anywhere.	F	5	4
#013	The system shall be able to act as a remote controller to the service-enabled objects.	F	5	5

Table 4.1: Early user requirements - Functional

[a]Customer Satisfaction Rating
[b]Customer Dissatisfaction Rating
[c]Functional requirement

Number	Description	Category	CSR[a]	CDR[b]
#002	The system shall run on a mobile device.	NF[c]-Operational	5	5
#009	The system shall handle the access of certain service-enabled objects.	NF-Security	2	1
#011	The system shall remember the previously seen objects.	NF-Performance	4	3
#012	The system shall have a minimalist/simple user interface.	NF-Look and Feel	5	5
#014	The system shall introduce different settings for novice and expert users.	NF-Usability	4	2
#015	The system shall provide supports for disable people.	NF-Usability	3	1
#016	The system shall be device-independent.	NF-Portability	5	4
#017	The system shall response fast.	NF-Performance	5	5
#018	The system shall be easy to use by every mobile device users.	NF-Usability	5	5

Table 4.2: Early user requirements - Non Functional

[a]Customer Satisfaction Rating
[b]Customer Dissatisfaction Rating
[c]Non Functional requirement

Chapter 5

Early Evaluation with Paper Prototypes

"As interfaces become ever more complex and development schedules seem to get shorter and shorter, you may find it useful to give up your user-interface modeling software for awhile in favor of something simpler. All you need is paper, pens, scissors, and your imagination."

—*Shawn Medero*

The preliminary user questionnaire presented in the previous chapter focused on the quantity of the feedback given regarding the early user interface design of UbiLens. Users, in this case participants, only gave feedback based on their imagination. Most of the times, this imagination could not be realised in the real life. This chapter deals with the paper prototype evaluation that allowed users to get hands on the application during the early stage of development. With paper prototyping, the second development cycle of the DIA Cycle was started (cf. 2.4.2—"Iterative and Incremental Development Cycle with Prototyping").

5.1 About Paper Prototyping

Improving feedback quality

Compared to the preliminary user questionnaire conducted previously, paper prototyping focused more on the quality of the feedback of the users. This was achieved by asking the users to actually try to interact with UbiLens instead of only imagining it. Even though it was in the form of paper, the paper prototype represented what the actual user interface would look like and behave. Nevertheless, with the length of time needed to conduct a single paper prototype evaluation, this evaluation could only be done with fewer participants than the preliminary user questionnaire.

The most effective usability evaluation

Snyder [2001] defines paper prototyping as a method of usability testing in which paper is used to design and test the user interface that is going to be developed. It has been proven to be the most effective and efficient way to test the usability of applications at the early stage of development process [Bolchini et al., 2009]. It is easy to create and does not need a lot of effort and time compared to higher fidelity prototypes (e.g. Flash) [Sefelin et al., 2003] It is also a suitable method to find out early what users actually want. If there are changes to the design or several features that the users do not even need them, the changes are easy to be integrated since the codes have not been written yet and it could save the coding time [Snyder, 2001].

Prerequisites

Before being evaluated, paper prototypes need to be designed and drawn. Everyone in the development team can take part in designing the paper prototype since no special technical skill is required [Bolchini et al., 2009]. During the evaluation session, two roles should be taken by the developers. One is the facilitator. He/she is the one who explains the tasks the users need to do and helps when users find any difficulties in doing the tasks. Another one is the *computer*. As the name implies, he/she, acting as a computer, simulates what the interface would do [Snyder, 2001].

5.2 UbiLens Paper Prototype Evaluation

The goal

The paper prototype evaluation was designed to evaluate the first prototype of UbiLens user interface and interaction methods [Reiners and Wibowo, 2009]. The evaluation focused on the naturalness of the interaction, the visual cues given, and the user interface in general. Furthermore, the evaluation aimed at digging other possible interactions which were more convenient to the participants.

5.2.1 Gathering Materials

The paper prototype

A set of paper prototype has been designed (cf. Figure 5.1). Materials, such as printing paper, card board, and post-it notes, were used. A model of an Android phone in real size (cf. Figure 5.1 - top left) was created to simulate the screen size and the physical buttons available on the mobile phone. This complies with the *Lessons from Early Stages Design of Mobile Applications* paper by de Sá and Carriço [2008]; which mentions that properties, such as weight, size, UI, and amount of data per screen, play important role in simulating the behaviour of users in the early stage of development process.

5.2.2 Evaluation Settings

Duration

The evaluation took approximately 30 - 60 minutes. This depended heavily on the understanding and curiosity of the participants. The less understand the participant, the longer it took to explain the tasks in details. The more curious the participant is, the longer it took to answer the questions asked by the participants.

Think-aloud

During the first few evaluation sessions, the participants were given a set of paper containing short description about the evaluation itself, tasks to do, and questions (cf. Appendix C—"Paper Prototype Evaluation Form"). Participants were asked to read all the tasks and questions by

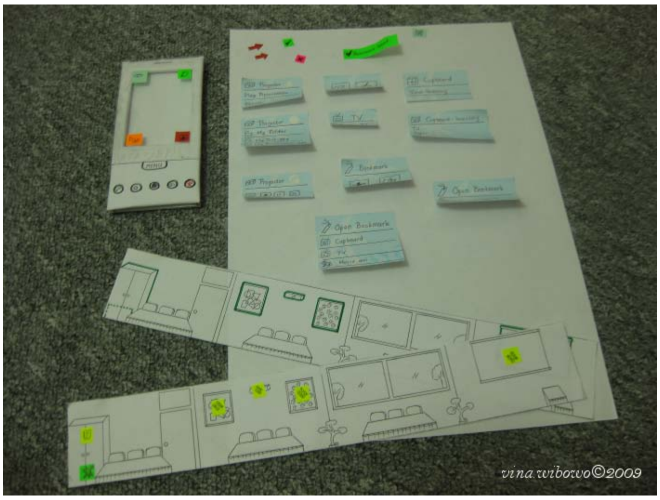


Figure 5.1: UbiLens paper prototype

themselves and say aloud what they were currently thinking [Wright and Monk, 1991].

The facilitator and the "computer"

There was also an observer who acted both as the facilitator, who observed the behavior of the participants, and the "computer" of the mobile phone. The observer only answered necessary questions. However, this method was proven not to be effective. Participants needed more explanation about the background of the application and why they had to do the tasks. Therefore afterwards, the evaluation method has been adapted so that the observer explained more details about the background of application and tasks. A suggestion also came from one participant that rather than giving participants alternatives or specific interactions that should be done, allowing participants to act independently given a specific goal would be better. Further adaptation of the evaluation method was made to fulfil this suggestion.

5.2.3 Participants Profile

Age

In total nine participants (four females and five males) were recruited to evaluate the paper prototype. The participants ranged between 24 - 36 years old with an average of nine years experience in using mobile phones. Several older candidates were asked whether they would like to participate in this evaluation. However, they tend to reject to help by saying that mobile phone is considered to be "young people's gadget"; even though they have been told that the user target for this application was everyone of any age.

Occupation

Appointments were made between the participants and the evaluator because of the considerably long time needed to complete this evaluation. Therefore, the participants were mainly from the social network of the evaluator. This, unfortunately, limits the profession of each participant; with six Master students (four with Media Informatics, one with Agriculture, and one with Food Chemistry major) and three computer scientists (one with psychology background). Nevertheless, four participants have never developed mobile applications before.

Left-handed

Additionally, one left-handed participant (male) was interviewed for only a short time due to lack of time. The reason behind this interview was to know how he, as a representative of left-handed people, interacted with his current mobile phone including the difficulties he met so far.

Figure 5.2 shows some impressions of the evaluation sessions.

5.2.4 Task Evaluation Result

Result summary

The result below is a summary gathered from the nine evaluation sessions held for each participant. The result is represented based on the tasks given during the evaluation process.

Task 1 - Searching for and selection of object

Giving presentation scenario

The participants were given with a scenario in which they were going to give a presentation in a room. Before beginning with the presentation, they needed to search for a projector and test the presentation slides there.

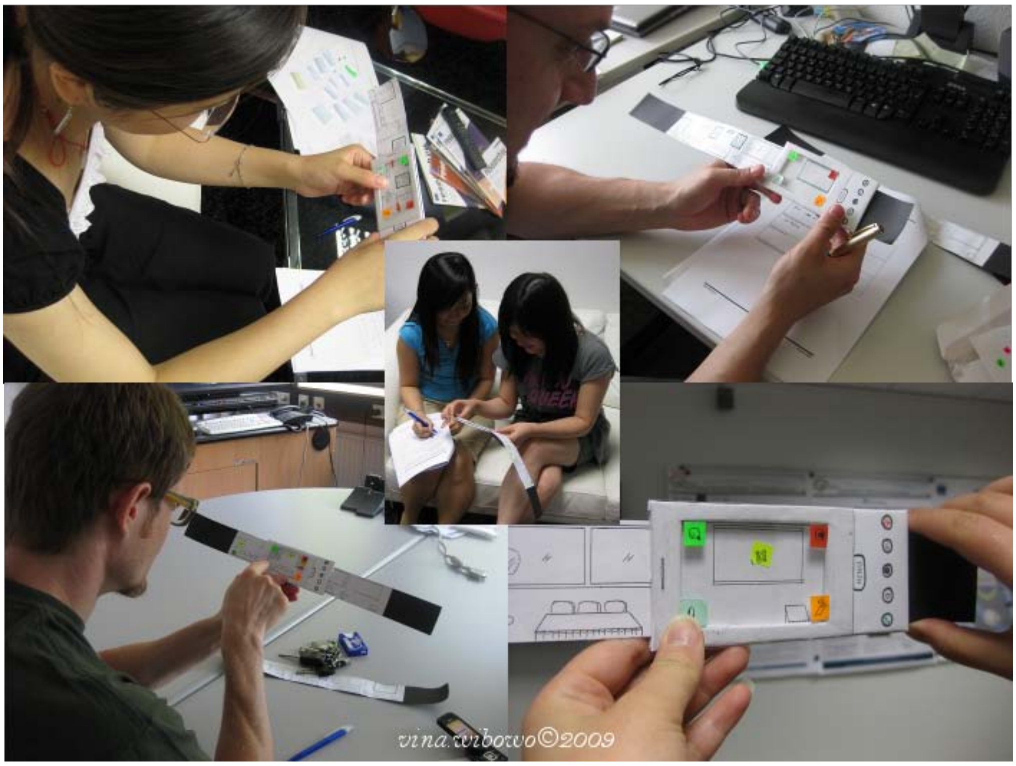


Figure 5.2: Paper prototype evaluation sessions

Smart Object Highlight

Simplified options

Three options to highlight smart objects seen through mobile phone's screen were given. These included a colour border, common icons, and a combination of both highlights. These options were different from the ones given during the questionnaire. The reason for this was to simplify the options.

Border still wins

Five out of nine participants said that having borders to highlight smart objects were preferred. One of the reasons was that the border was an intuitive way to highlight special or chosen items, such as in menus or file browser. Others said that having a border gave less distraction than having icons; because icons could occlude the object from view.

Additional icon increases the level of understanding

Four out of nine participants preferred having a combination of borders and icons. One participant said that the combination would be the first level to learn the interaction as it revealed not only that the object was smart but also the main function of the object itself. Another said that it could help common user to understand the feature of the object. However, as one noted, the selection of the icons need to be taken into consideration since the meaning of the icons should be clear.

None of the participants chose to have icons to highlight smart objects. One participant even mentioned that the user could identify the function of the object from the shape of the object itself. However, this argument was not entirely true since several smart objects, such as posters, might have different functions than its original physical functions.

Icons to identify functions

Out-of-View-Range Arrow

Arrows are displayed to give hints that there are nearby smart objects which cannot be seen through the screen. Previously, an option to have halos were given (cf. 4.2.3—"Visualisation of Smart Objects"). Since it was not as popular as the arrows, for the paper prototype evaluation, the option was omited. Instead, the participants were asked whether this feature (i.e arrow) helps them to find smart objects.

Arrows to indicate out-of-view objects

Seven participants found it useful to have the arrows while searching through smart objects. One participant mentioned that this mechanism was useful when there were not too many smart objects nearby. Otherwise, the application would always display arrows which were useless and could block the view. Another said that she did not notice the arrows at first but still said that they were useful.

Arrows were useful

One participant preferred not to have arrows to indicate nearby objects but maybe other mechanism, such as bright borders. One participant was not sure whether this arrow was of any help. She thought it would be better to get a list which let her know all the smart objects in the current view (i.e. screen).

Other suggestions

Smart Object Menu

All participants found the smart object menu was clear,

Simple menu

simple, and easy to use. One participant asked whether he should point to the smart object to control it (i.e. using menus) because to point to an object, one had to hold hands up and it was not convenient. He suggested switching off the camera and the users could retract their hands to further interact with the objects.

Task 2 - De-selection of smart object

Going back

At some point, when a smart object is chosen and the user is in the smart object's menu, the user wants to go back to the original screen where he/she can search for other smart objects. This process is called de-selection of smart object.

Escape buttons

Four out of nine participants were asked how they would do it. One of them imagined to have "back arrow" button, while the other preferred having "exit" button. Others would touch the area beyond the menu or click the menu title to search other objects.

Icon-only

Another question raised was whether an icon as a button was enough to indicate the feature or a special text, such as "search" or "scan", should be displayed along with the icon. Five participants said that icon was enough while the other four said more hints such as text to explain the meaning of the icon would be better.

Introducing gesture

Touching icon or button was offered for beginner or first time users, while gesture, such as shaking the phone, was offered for more expert users. The participants were offered with these options. Everyone liked the idea of shaking the phone since it was quick and easy. One participant, though, mentioned that this option could create misunderstanding for users who were used to other mobile devices, such as mp3 player, which mapped shaking as "change to next song".

Task 3 - Hidden objects

Interaction with occluded objects

In reality, several objects cannot be seen from one point of view because they are being occluded by other objects. This situation raised a question whether it is possible to interact with hidden objects.

Indicate hidden objects

The first thing to be taken into consideration is how hidden objects are highlighted. The highlight should be, on one hand, different with the visible object's highlight, while on the other hand, it should be similar to the visible object's highlight to maintain consistency.

Combination of border and icon

Three options were given to the participants: a dashed border, an icon with additional image, or a combination of both. Four participants agreed to have a dashed border as the highlight because it had a psychological effect that something was hidden. Only one participant wanted to have an icon with additional image which helped her to know the smart object's function while still indicating that the object was hidden. The other four participants preferred to have the combination of both because it was easier to distinguish and it presented clearer functions than either dashed border or icon only. Similar with the previous task (cf. 5.2.4—"Task 1 - Searching for and selection of object"), borders were still prefered to icons.

Task 4 - Drag&Drop

Drag&drop to exchange information

This task was to investigate whether the drag&drop action found in 2D desktop environment could be applied to 3D environment. In this project, drag&drop is used to exchange files or information between two or more smart objects.

Asking participants' opinions

Before the options on how to do this action were presented, four participants were asked how they would do it. Two participants said that they would touch the first object, move the phone until they found the second object, and then release their finger. Another participant proposed to have a "lock" menu which meant that the first object was chosen and locked for further interaction. Then he would move the phone and select the second object. Meanwhile, one participant had no idea how to do the drag&drop.

Drag&drop options

Two sets of steps were given in order to drag&drop in 3D environment. All participants were asked about the convenience of these steps and which set they preferred. First set would be to (1) focus the camera to the first object, (2) touch the first object for three seconds, (3) focus the camera to the second object, and then to (4) touch the projector canvas. Second set would be to (1) focus the camera to the first object, (2) drag the first object to the clipboard area, (3) focus the camera to the second object, and then to (4) drag the first object's icon on the clipboard to the second object.

Second set was prefered

Among nine participants, seven preferred to do steps in the second set as they were clearer, more natural, more intuitive, and more convenient. Only one participant said that the first set was better because the second one was not user-friendly. Another participant said that the combination between the first two steps on the first set and the last two steps on the second set would have been better.

Visual feedback

Drag&drop feedbacks

Visual feedbacks were given to indicate which smart object was "locked" or whether the transmission of files or information (drag&drop) succeeded. The participants were asked about the value of these visual feedbacks.

Feedback for dragging

The smart object's transparent icon is moving along following the touch of the users when the object is dragged. All participants had no problem with this feedback.

Feedback for dropping

After transmission, a visual feedback in the form of a check "✓" upon transmission success or a cross "x" upon transmission failure is displayed. Almost all participants grabbed the meaning of the check and cross symbol while one participant said that she needed an explanatory text for complementing the symbol.

Task 5 - Create bookmarks

How to create bookmarks

As pages of a book, smart objects can be bookmarked. The idea behind this bookmarking system is to allow users to interact with the smart objects from distance. This task required participants to create bookmarks by either touching the bookmark icon or dragging the smart object to the bookmark area. All participants did not have difficulties in performing both options. There was one remark from one of the participants that it would be better to have an add "+" symbol besides the smart object icon directly which meant "add to bookmark" while touching the bookmark symbol meant for opening bookmarks.

Task 6 - Open bookmarks

How to open bookmarks

Opening a bookmark can be done in two different locations. One is opening a bookmark directly on the phone using the phone's resources. Another would be to open on other smart object. All participants did not find any difficulties in following the instructions on the phone which were started by touching the bookmark icon. One participant suggested selecting the smart object first then touching the bookmark icon for opening bookmark on other smart object. Another suggestion was to have the chosen bookmark's icon displayed on the bookmark area so that it could be dragged multiple times.

Bookmark's look and order

Several questions were asked about how the presentation of the bookmarks looks like. The first question was whether the participants preferred a filtered list or an all-in-one list. Two participants preferred to have a filtered list based on the category of the smart objects. Others said a simple all-in-one list would not be a problem. Second question was how the bookmarks were ordered. One participant said the bookmarks should be ordered categorically while another said they should be ordered by time created. The others said that it did not matter. One suggestion came from one participant about the possibility to manage bookmarks as in desktop web browsers; which include renaming, arranging in folders, etc.

5.2.5 Icon Evaluation Result

Icon-based interaction

Icons play important roles in this application mainly because the limitation of mobile phone's screen space and users will mostly interact with icons to explore the features of the application. Therefore, a separate evaluation investigating the naturalness and intuitiveness of the icons to be used was done after the main evaluation mentioned above.

Application Icons

Common icons

Application icons define icons that are used throughout the whole application. Participants were given several options to choose from a collection of icons and free to give their own suggestions. Table 5.1—"Application icons" summarises the chosen icons.

Smart Object Category Icons

Icons by category

Categorical icons help users to determine the main function of the smart objects. Similar with application icons, participants were free to choose from a collection of icons or suggest their own for each smart object category. Table 5.2—"Smart object category icons" summaries the reception of offered icons.

Hidden Objects Icons

Star indicates hidden objects

The above icons are used to indicate the visible smart objects. Meanwhile, the offered solution for hidden objects was by adding additional image nearby the category icon. For example, this icon (cf. Figure 5.3) is to indicate that there is a hidden smart object which is categorised as audio object. A star is added to the original category icon. Three participants agreed with having a star while the others had other opinions. Two would like to have transparent icons; the other two would like to have the icons in greyscale; while the last two preferred to have shadows or dashed border surrounding the original icons.

Users will learn new icons

While one participant said that the selection of icons would

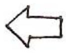
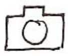
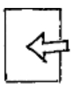
Description	Offered	Preferred by	Suggestions
Main icon		4	text with application name
Bookmark		6	 paper clip  star  mouse pointer
Clipboard		5	 memo board  pin  box  basket
Search		5	 back arrow  camera  binoculars
Exit application	   	2 2 0 5	-

Table 5.1: Application icons

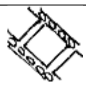
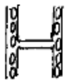
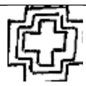
Description	Offered	Preferred by	Suggestions
Audio		9	-
Video	  	4 0 5	-
Audio & Video	  	4 0 2	separation between the audio and video icon (no overlapping) no need for this category
Info/Text		9	
e-Health		9	
Home appliance		7	this icon indicates power supply
Other/default		6	 blinking question mark  thunderbolt  exclamation mark  blinking exclamation mark  info

Table 5.2: Smart object category icons

Figure 5.3: Icon for hidden audio object

define the intuitiveness of the application, the other mentioned that whatever icons the developer chose, the users would learn and understand what they meant.

Placement of Main Menu Icons

Screen menu icons

The main menu is a set of four menus which are available throughout the entire application lifetime. Meaning that, these menus, represented by icons, will be displayed on the screen the entire time. The original arrangement is depicted in Figure 5.4.

Suggested icons arrangement

Nine participants were asked about how they felt about the menu arrangement on the screen. Five out of nine participants said that they liked the current placement and they did not express any difficulties in touching or dragging to or from the menus. Meanwhile, four participants had different opinions. One participant suggested that the exit menu is located at the bottom right corner, while the search and bookmark menu are positioned side-by-side on the top right corner of the screen (cf. Figure 5.5). The other three participants suggested locating the menus in one line (with the exit menu on the most right) either on the top or bottom of the screen.

5.2.6 Interview with the Left-handed

No support for left-handed

As mentioned before, a left-handed male was recruited to do a small interview about how he interacted with his current touch screen mobile phone. He said that left-handed people were normally forced to do the same interaction as right-handed people. This was due to the fact that, so far, the user interface on the mobile phone only supported right-handed people.

Figure 5.4: Original screen menu icons arrangement

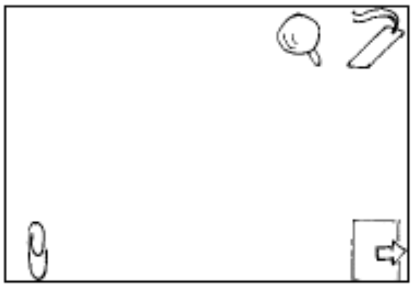


Figure 5.5: Suggested screen menu icons arrangement

Scrollbar problem

As one example, he mentioned about the scrollbar. The scrollbar is located on the right side of the screen while the text or list is located on the left side. Left-handed people, by nature, use their finger on their left hand to scroll the scrollbar. As a consequence, their left hand will occlude the text or the list that they are reading. In this case, the left-handed people are forced to use their finger on the right hand to control the scrollbar since there is no way to put the scrollbar on the left side of the screen.

Hardware problem

Another example also appears on the hardware of the phone. Take the HTC Dream, also known as T-Mobile G1. This phone has a slide keyboard which is opened from the left side of the phone. This means that the phone should be rotated counter-clockwise; resulting the buttons to be on the right side 5.6. Should there be any interactions with these buttons, right hand would be used.

Adaptable UI

One possible solution for this problem, if the mobile phone

Figure 5.6: Android G1 Phone

platform supports, is to mirror the user interface. For example, in this project, instead of having the exit menu on the top right corner, it is better to place the exit menu on the top left corner. However, this feature should be adaptable so that the right-handed people could still have the exit menu on the top right corner.

5.2.7 Contribution and Changes to Design

Contribution

These paper prototype evaluation sessions were conducted to get more ideas on how to design user-centred and user-friendly interface on mobile application. Based on the result of these sessions, the user interface design of this mobile application would be adapted accordingly. However, the adaptation did not depend entirely on this evaluation. Among other aspects, available technology and mobile platform capabilities and limitations also play important roles.

Design changes

The interaction and user interface design presented on this paper prototype evaluation would not be greatly changed in the next iterations. Several adaptations, such as drag&drop interaction selection, icon selections and consideration for left-handed people will be implemented in the next prototype.

Chapter 6

From Paper to Reality

"Kids saw their town from a different perspective, through the lens of a camera."

—Lisa Murray

With the feedback from the user's point of view, the third cycle of the DIA Cycle (cf. 2.4.2—"Iterative and Incremental Development Cycle with Prototyping") began. In this cycle, a prototype was designed, built, and evaluated to realise the concept of UbiLens. This chapter discusses how UbiLens was implemented.

6.1 System Architecture

Components of UbiLens system

UbiLens is a part of a larger system. It is installed on the users' mobile phone while the other components are located elsewhere. This system consists of five components: Smart Object, Smart Object Web Service, Object Recognition Component, Object Information Component, and UbiLens Mobile Application. Figure 6.1 depicts the relationships between these components.

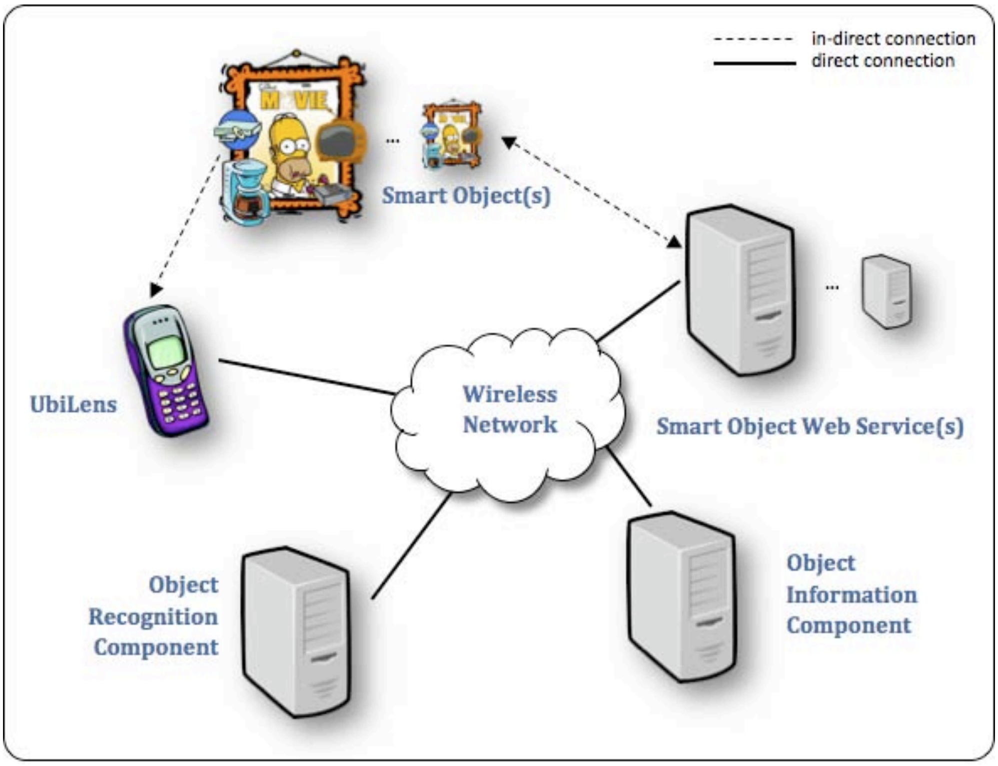


Figure 6.1: UbiLens system architecture

6.1.1 Data Transmission

Smart object discovery and visualisation

The components communicate with each other via wireless network connections. First of all, UbiLens sends the camera's image stream to the Object Recognition Component via socket connection. Upon recognition, the Object Recognition Component sends back the ID of the recognised object to the UbiLens mobile application along with the object's location on the camera image (i.e mobile phone's screen). Using this ID, UbiLens checks whether the object has been previously recognised and its properties have been stored. If not, UbiLens makes a request to the Object Information Component about the properties and services that the object offers. By using this information, either the one which is stored or acquired from the Object Information Component, UbiLens displays several visual cues that give hints to the users about the name of the object and its exact location on the mobile phone's screen.

Consuming service

The visual cues that are displayed on the screen can be touched for further interactions. When it is touched, UbiLens displays a menu containing a list of the smart object's services. From this list, users need to choose the service they want to consume. UbiLens then checks the type of the chosen service. Based on this type, UbiLens decides on which communication method it will execute to consume the service. For example, if the service type is *Servlet*, then UbiLens makes a call to the Servlet and waits for a reply. Further service types can be found in section 6.3—"Smart Object Web Service".

Drag&Drop

In Drag&Drop activity where two smart objects are involved, UbiLens requires the first object (i.e. the smart object of which service to be consumed) to be bookmarked first. In bookmarking there is no additional data transmission needed since UbiLens just basically stores the reference to the smart object along with all its services. For the second object (i.e. the object which consumes the service of the first object), the process of discovery and visualisation as described previously is repeated. After the second object is recognised, users drags the first object and drop it on top of the second on. UbiLens then checks the compatibility of both smart objects. When they are matched, UbiLens forwards the reference of the first object to the second one. It is then the responsibility of the second object to consume the service offered by the first object.

Example

In order to clearly picture how this data transmission between components works, consider the scenario presented in section 1.3.2—"Movie Day". Figure 6.2 depicts the sequence of data transmission when a user wants to play a movie trailer which is obtained from a movie poster. In this scenario, the user wants to play the movie trailer (a service) on TV (another smart object which also offers services). It is essential that before the information exchange between smart objects is possible, a reference to one of the smart objects needs to be stored (cf. section 6.6.3—"Bookmarking"). Then to recognise the second object, UbiLens repeats the same process above (without bookmarking). UbiLens, forwarding a reference to MovieTrailerWebService, tells the TVWebService to play the video offered by the MovieTrailer-

WebService. TVWebService then requests the movie trailer to the MovieTrailerWebService. After the video, either in the form of file or reference, is obtained, TVWebService tells its corresponding TV smart object to play it.

6.2 Smart Object

What are smart objects?

Smart objects are ordinary everyday objects that have been tagged with their digital representation in the form of web services. One example would be a movie poster that has been tagged with a service to book a ticket or to watch the trailer. By using UbiLens, users can directly interact with the movie poster to consume its services.

Object description

In the UbiLens system, each smart object is represented with an XML object description. This description is organised by Object Information Component (cf. section 6.5—"Object Information Component"). An example of the XML description can be seen in figure 6.3. It contains the smart object properties, such as ID and name, and the services it offers.

TakeAway

TakeAway property defines whether a smart object can be bookmarked. Bookmarked items are remotely accessible. Users can control the smart object elsewhere. However, it does not make sense to bookmark all smart objects. Therefore, each smart object needs to define whether a bookmark is feasible. Further information regarding bookmarking can be read in section 6.6.3—"Bookmarking".

6.3 Smart Object Web Service

1 to n services

A smart object offers more than one service. These services can either be located at the same or a different location.

Service description contents

An object description of each service can be found in the XML description (cf. Figure 6.3). It consists of a description, a type, and a value. The description describes what the service does. The type defines the type of the service. The

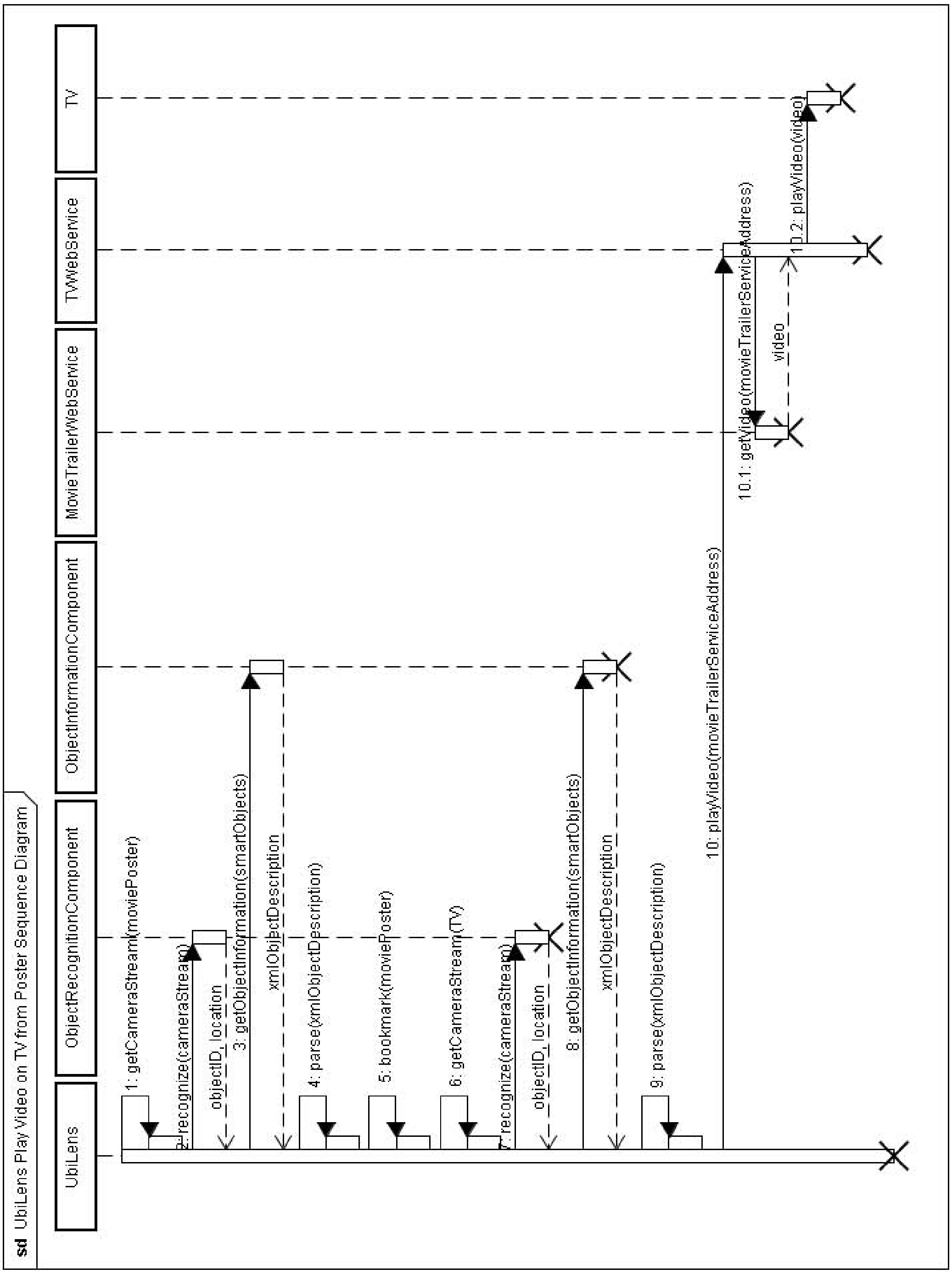


Figure 6.2: Play video from poster sequence diagram

```
- <SmartObjectIdentificationResponse>
  - <SmartObject>
      <Id>7</Id>
      <Name>Human Computer Interaction</Name>
      <Icon>book</Icon>
      <Visible>1</Visible>
      <TakeAway>1</TakeAway>
    - <Services>
      - <Service>
          <Description>Get information</Description>
          <Type>URL</Type>
          <Value>http://www.hcibook.com/e3/</Value>
        </Service>
      - <Service>
          <Description>Buy now</Description>
          <Type>URL</Type>
        - <Value>
            http://www.amazon.com/Human-Computer-Interaction-3rd-Alan-Dix/dp/0130461091
          </Value>
        </Service>
      </Services>
    </SmartObject>
  </SmartObjectIdentificationResponse>
```


Figure 6.3: XML object description example

value contains the address or reference to the corresponding service.

service types

The service type is used by UbiLens to decide which communication or interaction procedure needs to be taken to consume the service. During the implementation phase, six service types have been defined. They are *Audio* (i.e. audio file), *AudioPlayback* (i.e. audio player), *Servlet*, *URL*, *Video* (i.e. video file), and *VideoPlayback* (i.e. video player). Of course, these types are not finalised yet. They were defined just to facilitate the first prototype of the system. Additional or updated types can be added or modified to accommodate future services.

6.4 Object Recognition Component

Recognition mechanism

Before interaction with smart objects is possible, smart objects need to be recognised first. For the first prototype of

UbiLens, image recognition algorithm (cf. section 3.3.2—"Image Recognition") was chosen to recognise smart objects. However, other object recognition such as barcode, (indoor) location, infrared, RFID, and Bluetooth recognition can also be used. Further information concerning the choice of recognition mechanism can be found in section 3.3—"Object Recognition".

How it works

Image recognition algorithm used in this system was based on Scale-invariant feature transform (SIFT) and developed by OFFIS[1]. It is installed on a dedicated server and works with client-server manner. All the client needs to do is to send a greyscaled camera's image stream to the server. Applying the SIFT algorithm, the server recognises the object found on the image frame and sends back the corresponding object ID to the client. At the implementation phase of this project, the image recognition algorithm was only capable of recognising one object per image frame.

Lots of pictures are needed

A set of pictures of every angle of the object is required for the recognition process. Before the image recognition server is started, the pictures need to be stored on the server. While starting, the server will scan each of the pictures and put it in its internal database. The more pictures collected, the better the recognition process is. However, more efforts are needed to take all the pictures. This also means that upon start up, the server needs to scan a lot of pictures, which may take time. Nevertheless, this only happens once for every server start up.

Marker-less recognition

The "no need for physical marker" property of the image recognition mechanism (cf. section 3.3.2—"Image Recognition") does not spoil the appearance of the object itself. Imagine having a smart TV with a large barcode printed on it.

Client-server advantages and disadvantage

The client-server manner employed by the system makes recognition process faster and modification of the picture database more convenient compared to recognition done on the mobile device, which has limited processing power and storage. Not only this, but also the changes of the image recognition algorithm do not lead to re-implementation and

[1] http://www.offis.de/

re-installation of the mobile device (i.e. client). Nevertheless, a fast and responsive network connection is needed to make sure a fluid recognition process.

Image recognition problems

The use of image recognition algorithm to recognise smart objects cannot escape from several problems. One huge problem is lighting. Too bright or low-lit rooms makes smart objects difficult to be recognised or worst not to be recognised at all. Additionally, at the beginning of this thesis work, indicating hidden or occluded objects is one of the research questions to be answered (cf. section 2.3—"Research Questions"). By using image recognition mechanism it is not possible to tell whether a smart object is hidden or occluded because if it is, the server will not be able to recognise the object. Hence a workaround has been done to be able to evaluate and answer this question. Instead of recognising the smart object itself, the image recognition algorithm recognised the object that is placed in front of the smart object. This workaround is not valid for real application since it is not dynamic. Another question that may occur is what if the smart object is occluded by another object. Further discussion regarding this problem can be found in section 8.2—"View of the Future".

6.5 Object Information Component

XML descriptions database

As mentioned before, each smart object has its own XML object description which contains its properties and services offered (cf. Figure 6.3). The Object Information Component is responsible for the coordination of these XML descriptions. The component acts as a database which stores all smart object XML description files and forwards them to any client.

Decision to request

When a smart object is recognised by the Object Recognition Component, UbiLens mobile application checks its internal storage whether the object has been recognised before. If it has not been recognised, UbiLens requests the Object Information Component to forward the corresponding XML description. However if the object has been recognised before, the properties and the services that the smart object offers are already stored inside the UbiLens. Therefore, no additional request is needed and thus some bandwidth are saved.

6.6 UbiLens Mobile Application

UbiLens in the form of mobile application is the main focus of this thesis project. Developed and implemented on Android platform, UbiLens mediates the interaction between the users and service-attached real world objects. Not only that UbiLens gives cues which objects are smart, but it also makes it easy for the users to consume services offered by the objects.

6.6.1 Android as the Base

Android's development

Released in October 2008 by Google and the Open Handset Alliance [Burnette, 2008], Android has become one of the leading mobile operating systems in the world. In February 2010, 60,000 Android handsets were shipped every day [Kumparak, 2010] and this number kept increasing. Who knows that later in the near future all mobile phones will be based on Android platform.

Android's speciality

So what is so special about Android? First of all, it is open source. Don't all people love open source where everything is more or less free? It offers software toolkit for mobile phone which can be coded with Java, instead of a completely new programming language. Last but not least, it is available on many handsets which do not come from the same manufacturer [Burnette, 2008].

Why Android?

The reasons above made the decision why this thesis project was based on Android. Furthermore, it considered the separation between the business logic and the UI codes. Developers can alter the UI without recompiling the whole codes. Additionally, Android supports tasks delegation, in which certain tasks, like viewing contacts list or opening a website, do not need to be handled by the current application.

The application only needs to say "I want to open this website" and the other applications who can open a website, in this case the Browser, will answer to this call and open the corresponding website. This reduces the developer's effort to expand their application's features.

6.6.2 Visualisation of Real World Objects

Visualisation plays an important role in UbiLens. It displays certain cues to help users find and interact with smart objects easily.

Layers in UbiLens UI

The UbiLens' user interface is divided into three layers. They are *Camera View* layer, *UbiLens View* layer, and *Drag Drop* layer, which is divided into two sub-layers. These layers were defined so that each layer is independent of the other layer. This ensures the fluidness of the user interface. Figure 6.4 summarises the UI layers of UbiLens.

Camera View

Camera View layer is located at the very bottom of the UbiLens' user interface. It displays the live camera image stream of the mobile phone. Figure 6.5 shows the UbiLens' Camera View layer.

UbiLens View

On top of the Camera View layer is the *UbiLens View* layer. It is responsible for giving visual cues to the users whenever a smart object is spotted. Using the object ID achieved from the object identification component and the object properties from the object information component, UbiLens draws the name and the picture (icon) of the corresponding object at the location where the object is seen on the mobile phone's screen.

Frame border

Additionally, a yellow frame border is drawn surrounding the phone's screen. This frame was a compromised made to realise one of the requirements mentioned during the preliminary user questionnaire (see figure 4.6(a)). According to the questionnaire, the frame should border the object. Nevertheless, the currently used recognition algorithm is only capable of returning the ID of the object along with its location. Figure 6.6 shows how the visual cues on the UbiLens View look like.

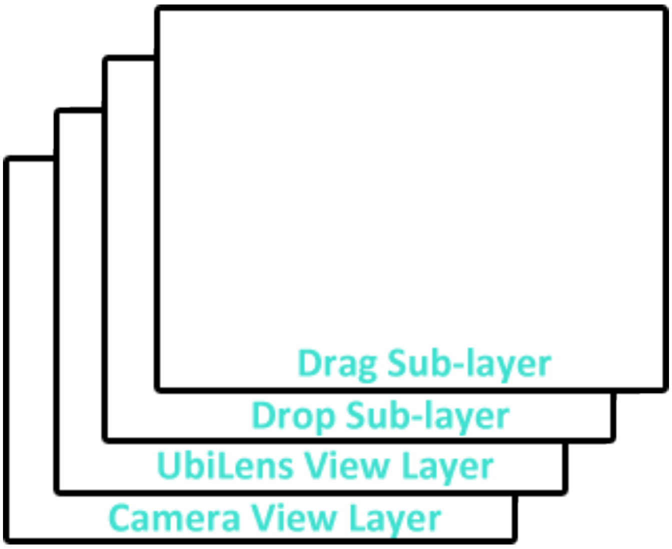


Figure 6.4: Layers in UbiLens' UI

Figure 6.5: Camera View layer

Hidden objects

Sometimes a smart object can not be seen from the users' point of view because it is occluded by other objects. Therefore, two different sets of visual cues are given. One set is for the visible smart object and the other is for smart object which is currently hidden or occluded by other object. These help users determining whether the smart object is the one in front of them or the one behind it.

Drag Drop layer

The next in line is the *Drag Drop* layer. This layer is later divided into two sub-layers: *Drag* sub-layer and *Drop* sub-layer.

Figure 6.6: UbiLens View layer: (a) visible and (b) hidden smart objects

Drag sub-layer

The Drag sub-layer, which covers the whole screen, is located on top of Drop sub-layer. It gives users visual feedback for the Drag&Drop interaction. Each time a user is dragging an object, a corresponding *ghost* is displayed to guide the user visualise the path of the dragging motion. See figure 6.7 for better visualisation. Further details concerning Drag&Drop can be seen in section 6.6.3—"Drag&Drop".

Drop sub-layer

The Drop sub-layer consists of two areas: *First Drop* and *Second Drop* area. These areas indicate two different sections in the Drop layer which play important roles in Drag&Drop action. Figure 6.8 depicts the placement of areas in the current prototype of UbiLens. The red polygons frame the First Drop area while the rest is the Second Drop area. The area does not need to be in the form of a perfect rectangle. Figure 6.8 (a) shows only a part of the First Drop area. Of course, this placement could be alternated.

6.6.3 Interaction with Real World Objects

Not only UbiLens is able to find and tell users about smart objects in the vicinity, it also bridges the interaction between the users and the services that smart objects offer.

Figure 6.7: Dragging an object

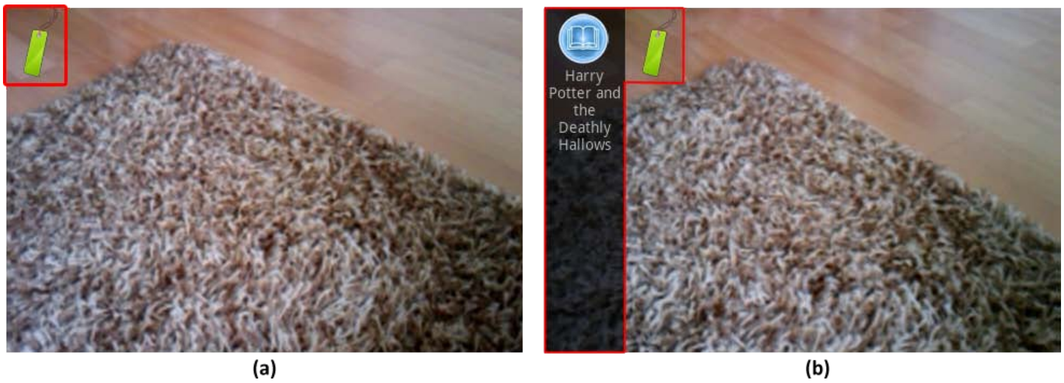


Figure 6.8: Drop sub-layer: (a) Hidden First Drop area and (b) Full First Drop area

Searching for Smart Objects

Step 1: finding the object

Before any interaction is possible, finding the objects in question is the most important thing. Using UbiLens, users only need to pan the mobile phone's camera throughout the whole room. When any smart object is found, UbiLens tells the users by displaying the corresponding visual cues (see figure 6.6. This technique has been kept from the beginning of this thesis project.

Visual cues delays

Delays between the time when a user pans his/her mobile phone's camera and the time the visual cues shown happen sometimes. The delays made the movement of the icon/name of the smart object displayed on the screen appears to be jumping. These delays were due to the delays on the wireless connection. The faster the wireless router connecting UbiLens system components is, the more fluid the displayed visual cues are.

Smart Object Organiser

In order to reduce the delays and further reduce the needs to connect with the Object Information Component as it costs bandwidth and money, UbiLens introduces a so-called *Smart Object Organiser*. This component, located inside the UbiLens application itself, organises the need to connect with the Object Information Component. As mentioned previously, the visual cues displayed on the screen indicating the smart object are composed from the information received from the Object Information Component. When an object is recognised by the Object Identification Component, the ID of the object is stored in a list maintained by the Smart Object Organiser. The organiser checks whether the ID has already been in the list. If no, a request is made to the Object Information Component asking the properties of the object. If yes, no request is made and the information stored in the list is used to generate the visual cues. Hence, the visual cues will appear quicker and some bandwidth will be saved.

Interaction using Menus

Dialog box menus

Menu, in the form of a dialog box, is a very common form of user interface on mobile devices. According to the questionnaire done previously (cf. 4.2.3—"Visualisation of Smart Object Functions"), menus were more expected than fancy gestures. Answering to the users' request, UbiLens displays a simple menu containing all services that the chosen smart object offers. In order to trigger the choosing of the smart object, users only need to touch the corresponding smart object's icon/name. A menu like the one shown in figure 6.9 will be displayed. To further interact with the menu, users do not need to pan the mobile phone on the smart object. They can retract their hands and hold the phone closer.

Figure 6.9: Smart object's menu

The smart object menu consists of a header with smart object's icon and name and a list of all the offered services. This list is generated automatically based on the XML object description shown in figure **??**. In this case (cf. Figure 6.9), users will be forwarded to the corresponding websites whenever one of the service is chosen.

Menu generated from XML object description

Bookmarking

Users are able to interact with the smart objects on the spot. Meaning that the interactions are mostly done in front of the corresponding object. There are times when interaction should also be possible elsewhere. Consider a smart object in the form of a book. Let say a user finds the book at their friend's house. At that time they do not want to purchase the book but maybe later. How can this be done?

The importance of bookmarking

UbiLens allows users to bookmark the found smart objects for later use. Users only need to touch the bookmark button which is available on the smart object's menu (see figure 6.9). More advanced way is to directly touch the icon/name of the smart object, drag it, and drop it (cf. Figure 6.10) on top of the bookmark drawer (see figure 6.8; the bookmark drawer is the area which is surrounded by the red polygons). Bookmarked items can be consumed by tapping the corresponding bookmark icon located in the bookmark drawer.

Consuming services elsewhere

Figure 6.10: Add bookmark by dragging smart object to bookmark drawer

Not all smart objects can be bookmarked

Of course, not every smart object can be bookmarked. Take example of a coffee machine. It offers services to make coffee and cappuccino. In this case, it does not make sense to bookmark the coffee machine. Imagine that a user does bookmark the coffee machine and the user consumes the make coffee service just before they arrive at their house. What if there is no cup that will catch the coffee made? In order to avoid this problem, in the XML object description, a smart object states whether it can be bookmarked (see figure **??**, TakeAway tag). This property is then also reflected on the menu of the smart object (cf. Figure 6.11). Notice the bookmark button is missing.

Drag&Drop

Limitation of consuming services on mobile phone

At some point, consuming services offered by the smart object on a mobile phone is not convenient to do. This holds especially for services that require certain level of qualities, such as a service to watch a movie trailer or listen to a sample music. With the limitation of the screen space and audio quality, consuming these services in other place is preferable.

Information exchanges

UbiLens lets information to be exchanged between two smart objects. This is called Drag&Drop, meaning that information is dragged from one smart object and dropped to another smart object. The first smart object (i.e. the *provider* object) is the smart object of which services shall be con-

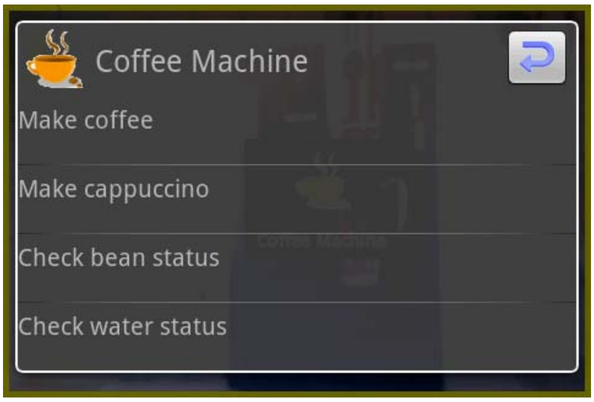


Figure 6.11: Smart object's menu without bookmark

sumed by the second smart object (i.e. the *consumer* object). And not the other way around. Drag&Drop should not to be confused with the usual drag (i.e. to touch and move object on the phone's screen) and drop activity (i.e. to release the dragged item on top of a certain area).

How to Drag&Drop

Drag&Drop begins with *locking* the provider smart object. In the current implementation of UbiLens, this is done by creating a bookmark of the smart object (cf. section 6.6.3—"Bookmarking"). Then, users need to search for the consumer smart object by panning the mobile phone's camera until the desired smart object is spotted on the screen. While still displaying the icon/name of the consumer smart object, users hold the bookmark entry of the provider smart object, drag, and drop it to the location of the consumer smart object. When the type of the services match, UbiLens forwards the reference of the provider smart object to the consumer smart object. Hence, the consumer smart object can consume the service offered by the provider smart object. At the moment, UbiLens supports Audio service to be dropped on to a smart object with AudioPlayback service while Video to VideoPlayback. See section 6.3—"Smart Object Web Service" for types of services.

Compatibility cues

When the bookmark entry of the provider smart object is being dragged onto the consumer smart object, UbiLens gives users visual cues to indicate whether both objects are compatible with each other. A green border frames the mobile phone's screen to indicate that the consumer smart object is able to consume the services offered by the provider smart object (cf. Figure 6.12(a)). A red border tells otherwise (cf. Figure 6.12(b)).

Choosing the provider and the consumer services

A smart object can offer more than one services. In Drag&Drop two kinds of services are involved: the *provider* service and the *consumer* service. The provider service is one service among all the services offered by the provider smart object and the consumer service by the consumer smart object. UbiLens needs to decide which service to be the provider and which to be the consumer. The services belong to a smart object is stored in a list. UbiLens iterates through this list and decides which service to be either the provider or the consumer by looking at the service type. Thus, in order to make UbiLens choose one service over the other services, the order of the services in the XML object description needs to be taken care of. The more important the service is, the earlier the service is mentioned in the XML object description.

Example

Let's take example on a movie poster smart object which offers services to watch movie trailer, to buy the movie's DVD, or to rent the DVD (cf. Figure 6.13). This smart object acts as the provider. UbiLens iterates through the smart object service list and decides which service has type Audio or Video, the currently supported provider service types. The choice then goes to the watch movie trailer service. The user chooses the consumer smart object to be a TV which has services to playback a video and to get TV programme. UbiLens, by looking at the service type, decides to choose the playback video service of which service type is VideoPlayback. When the movie bookmark is dropped on to the TV, the TV will then play the corresponding movie trailer.

6.7 A Spin-off: Hydragizer

energy efficiency

Hydragizer was developed within the third cycle of this thesis project. It was a part of Hydra middleware[2] demonstra-

[2]http://www.hydramiddleware.eu

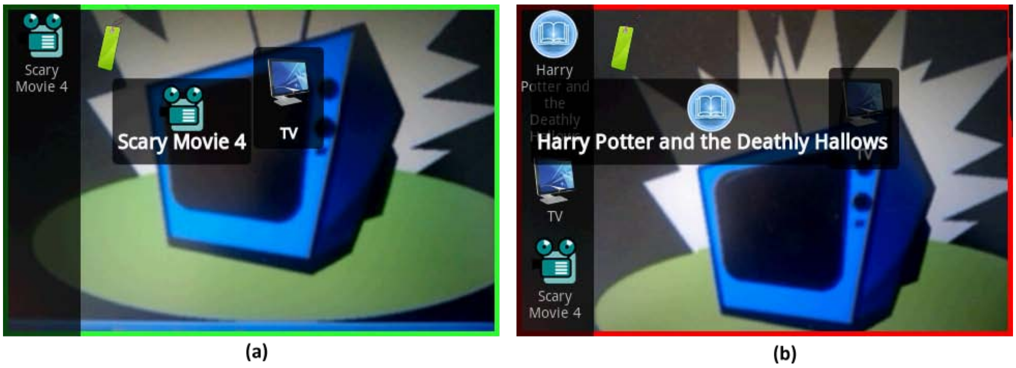


Figure 6.12: Compatibility of Drag&Drop objects: (a) green border indicates both objects are compatible, while (b) red border indicates otherwise

tor which was exhibited at CeBIT 2010 in Hannover, Germany. The demonstrator focused on raising awareness of energy efficiency. It was such a huge success in grabbing attention of potential customers (i.e. business customers and end-users) and mass media.

What Hydragizer does

Hydragizer adopts the visualisation and bookmarking techniques of UbiLens. Using the same architecture as UbiLens, with some additional components, Hydragizer visualises the current consumption and the yearly price of the smart object spotted on the camera image. The bookmarking technique is used to send the smart object to a central monitoring display. To get better idea on how Hydragizer works, watch the project video online[3] . This spin-off proves the extensibility of UbiLens.

[3] http://www.youtube.com/watch?v=pipapGP1Lvs

```
- <SmartObjectIdentificationResponse>
  - <SmartObject>
      <Id>11</Id>
      <Name>Scary Movie 4</Name>
      <Icon>movie</Icon>
      <Visible>1</Visible>
      <TakeAway>1</TakeAway>
    - <Services>
      - <Service>
          <Description>Watch trailer</Description>
          <Type>Video</Type>
          <Value>http://www.youtube.com/watch?v=h0zAlXr1UOs</Value>
        </Service>
      - <Service>
          <Description>Buy DVD</Description>
          <Type>URL</Type>
        - <Value>
            http://www.amazon.de/Scary-Movie-4-Anna-Faris/dp/B000FMH844
          </Value>
        </Service>
      - <Service>
          <Description>Rent</Description>
          <Type>URL</Type>
          <Value>http://www.lovefilm.de/film/Scary-Movie-4/57895/</Value>
        </Service>
      </Services>
    </SmartObject>
  </SmartObjectIdentificationResponse>
```


Figure 6.13: XML object description of a movie poster

```
– <SmartObjectIdentificationResponse>
  – <SmartObject>
      <Id>1</Id>
      <Name>TV</Name>
      <Icon>display</Icon>
      <Visible>1</Visible>
      <TakeAway>1</TakeAway>
    – <Services>
      – <Service>
          <Description>Video playback</Description>
          <Type>VideoPlayback</Type>
          <Value>http://192.168.0.1:8000/VideoPlayerService</Value>
        </Service>
      – <Service>
          <Description>Get TV programme</Description>
          <Type>URL</Type>
          <Value>http://www.tvinfo.de/</Value>
        </Service>
      </Services>
    </SmartObject>
  </SmartObjectIdentificationResponse>
```


Figure 6.14: XML object description of a TV

Chapter 7

Time to Evaluate

"Usability is like oxygen - you never notice it until it is missing."

—Anonymous

Evaluating a product before shipping it to the market is an important step to ensure the success of the product. Bugs, usability problems, unnecessary features can be found through several evaluation sessions. This chapter presents the evaluation sessions that have been conducted in order to assess UbiLens.

7.1 The Importance of Evaluation

Goals of evaluation

According to Dix et al. [2004], the main goal of an evaluation is to ensure the behaviour of a certain product meets the users' expectations and requirements. Furthermore, it is also effective to find specific problems within the system and assess users' experience in interacting with the product.

UbiLens acceptance

Align with the goals mentioned above, this evaluation was conducted mostly to measure the acceptance of UbiLens in bridging the interaction between users, real world objects, and their services. In this evaluation, the difference between non-tagged and tagged environment was also investigated.

Further details about these environments will be described in the next section.

7.2 Evaluation Settings

Before the evaluation began, several settings must be planned out in order to achieve the evaluation goals. This section describes these evaluation settings in details.

7.2.1 Two User Groups

Division into groups

In this evaluation, the participants were divided into two user groups. Half of the participants were placed in the first user group and the other half in the second user group. The decision to place a certain individual in a certain group was alternated, namely, the first, third, fifth, etc participants went to the first group while the second, fourth, sixth, and so on participants went the the second one.

7.2.2 Tasks to Do

Lottery scenario

A scenario has been compiled in order to guide the participants completing several tasks in this evaluation. The story starts with the participant, as a lottery winner, receiving a new high-tech apartment. Together with the representative from the lottery company, who is actually the evaluator, they enter the apartment for the first time. Inside the apartment, several objects are considered to be smart. They are told that smart objects offer services beyond their usual functions. They then use UbiLens to discover and to interact with the smart objects.

Participant's tasks

Each participant was given six main tasks, including searching for smart objects, checking out smart objects' services, bookmarking smart objects, opening bookmarks, searching for smart objects in another room, and Drag&Drop. Each task consists of several small sub-tasks. A complete listing of the tasks can be found in Appendix D—"User Evaluation Tasks".

Switch environments

Among the six main tasks handed to the participants in the first group, four of them were done in a non-tagged environment and the other two in a tagged environment. The second group did the opposite. This was to ensure that the participants from both groups have the same experience to try two different environments and give their opinions.

Tasks on index cards

Based on several experiences in observing user evaluations, when participants were given all the task descriptions at once, they tended to frequently glance at the next task even though they have not done dealing with the current one. This could spoil the result of the evaluation because sometimes the next task gives hints on how the previous task is solved. Keeping these in mind, during this thesis project's evaluation sessions, each task was written on a separate index card. Participants had to read them one by one. Meaning that the next card could only be read when the previous task on the previous card has been completed.

7.2.3 Non-tagged vs. Tagged Environments

Two rooms = two environments

Two rooms, or rather one big room divided into two rooms, were prepared. One room simulates a non-tagged environment and the other simulates a tagged environment. Both environments were set in a way that the participants would not gain any learning process from the previous room. For example, a postcard in one room is smart while another postcard in the other room is not smart. Furthermore, the rooms were designed so that they looked like ordinary living rooms and did not look like controlled labs.

Non-tagged environment

The first room has a dimension of four metres by four metres. Several objects, both smart and non-smart objects, were placed randomly inside the room. There were no additional tag printed on the smart objects. With naked eyes, these smart objects looked exactly like any other ordinary objects. Table 7.1—"List of non-tagged smart objects" summarises the smart objects found in the first room along with their services. Several non-smart objects were also placed inside the room. These include couches, lamps, CDs, an extra book, and an extra picture. Figure 7.1 illustrates the room simulating the non-tagged environment.

Object	Visibility	Services	
		Description	Type
TV	Visible	Video Playback	VideoPlayback
Paper Lamp	Visible	Get current consumption	Servlet
		Get yearly price	Servlet
Human Computer Interaction (book)	Visible	Get information	URL
		Buy now	URL
Bali (postcard)	Hidden	Get Wikipedia article	URL
		Book a flight	URL
Hydra Middleware (poster)	Visible	Watch video	URL
		Go to project page	URL
Coffee Machine A	Visible	Make coffee	Servlet
		Make cappuccino	Servlet
		Check coffee bean status	Servlet
		Buy coffee bean	Servlet
		Check water status	Servlet
Vina	Visible	Go to Vina's blog	URL
		Go to Vina's Facebook page	URL

Table 7.1: List of non-tagged smart objects

Tagged environment

The second room measures three metres by four metres. Similar to the first room, several objects, both smart and non-smart ones, were scattered inside the room. Some could be found on a table and some on the wall. However, all smart objects in this room had QR codes (cf. 3.3.1—"Physical Marker" printed on top of each object. The non-smart objects did not have any QR codes. A complete listing of the smart objects found in the second room can be seen in Table 7.2—"List of tagged smart objects". Figure 7.2 presents the view of the second room.

7.2.4 Think Aloud

What is *think aloud*

Each participant was asked to talk through what they were

Figure 7.1: Non-tagged environment

Object	Visibility	Services	
		Description	**Type**
HiFi	Visible	Audio Playback	AudioPlayback
DVD Player	Hidden	Get current consumption	Servlet
		Get yearly price	Servlet
Play Station 3	Visible	Get current consumption	Servlet
		Get yearly price	Servlet
A Zoological Garden (painting)	Visible	Get painter information	URL
		Buy as a poster	URL
Bubbly (music CD)	Visible	Listen to sample music	Audio
		Buy song	URL
		Get Wikipedia article	URL
Coffee Machine B	Visible	Make coffee	Servlet
		Make cappuccino	Servlet
		Check coffee bean status	Servlet
		Buy coffee bean	Servlet
		Check water status	Servlet

Table 7.2: List of tagged smart objects

Figure 7.2: Tagged environment

currently doing. This includes what happened with the system when they did something to the user interface, why they did that, and what they were trying to do [Dix et al., 2004]. This method, called *think aloud*, is simple yet effective to capture how exactly the participants handle the application. With this, usability problems can easily be found since the participants tell what went wrong right after it happened.

Co-operative evaluation

A subset of think aloud method, called *co-operative evaluation* was also used in this evaluation to encourage the participants to criticise the system [Dix et al., 2004]. This was done by asking the participants rather than waiting the participants to tell the evaluator about a certain issue. Even though at the beginning of the evaluation the participants were reminded about the think aloud method, they often forgot or even were a bit shy to express their thoughts. That is why co-operative evaluation method was proven to be more effective than the original think aloud method.

The evaluator captured the participants' thoughts by writing them down with paper and pencil. A very fast writing and capturing skill was needed. An extra voice recorder were also used to support the written points. Sometimes when the participants talked real fast, it was almost impossible to write them down. However, the participants were told to remember these points and later on write them down on the prepared questionnaire forms.

Capturing thoughts

7.2.5 Measurements Taken

Several measurements were gathered to evaluate the speed of the system and the interaction between the system and the participants. Each sub-tasks were measured based on time. During both searching for smart objects and searching for smart objects in another room, the evaluator noted the time and the number of found smart objects.

Things to measure

7.3 Location and Duration

The evaluation sessions were conducted at Fraunhofer Institute, Sankt Augustin (near Bonn), Germany from 19 - 23 April 2010. Each session took approximately 30 - 45 minutes. It mostly depended on the interactivity of the participant. The more active the participants were in trying out some features outside the tasks given, the more time they needed to complete all the tasks.

1 week evaluation

7.4 Participants Profile

In total, 22 people were participated in this evaluation. They were 19 males and 3 females. Their age ranges between 22 - 36 years old, with an average of 28. The distribution of age can be seen in Figure 7.3. In average, they have been using mobile phones for 9 years.

Participants' Bio

Since the location of the institute, where the evaluation ses-

Location issue

sions were conducted, is far from Bonn city centre, it was difficult to invite people from outside the institute. Therefore, all participants, except one, have affiliation with the institute. Almost half of the participants were students and more than a quarter were scientific researchers. Figure 7.4 illustrates the distribution of occupation among the participants.

IT background

18 out of 22 participants have background in Information Technology or Computer Science. However, only 10 people have developed mobile phone applications prior to the evaluation session.

Figure 7.5 depicts several participants during evaluation sessions.

7.5 The Outcome

Two results

The evaluation result is divided into two parts. One is the result based on the performance of the participants in completing the given tasks. The other is the result of the questionnaire given to measure the satisfactory level in using UbiLens.

7.5.1 Task-based Evaluation

What to measure

In task-based evaluation, the participants were given a set of tasks to be completed. The evaluator noted down the time needed to perform the task and also the behaviour of the participants in dealing with the tasks. Comments, thoughts, and complaints from the participants as well as problems encountered were also recorded.

Task 1 & 6: Searching for Smart Objects

Two searching sub-tasks

In these tasks, the participants were asked to search for smart object in the room they are currently in. At first, they had to search for smart objects by pure guessing. Then,

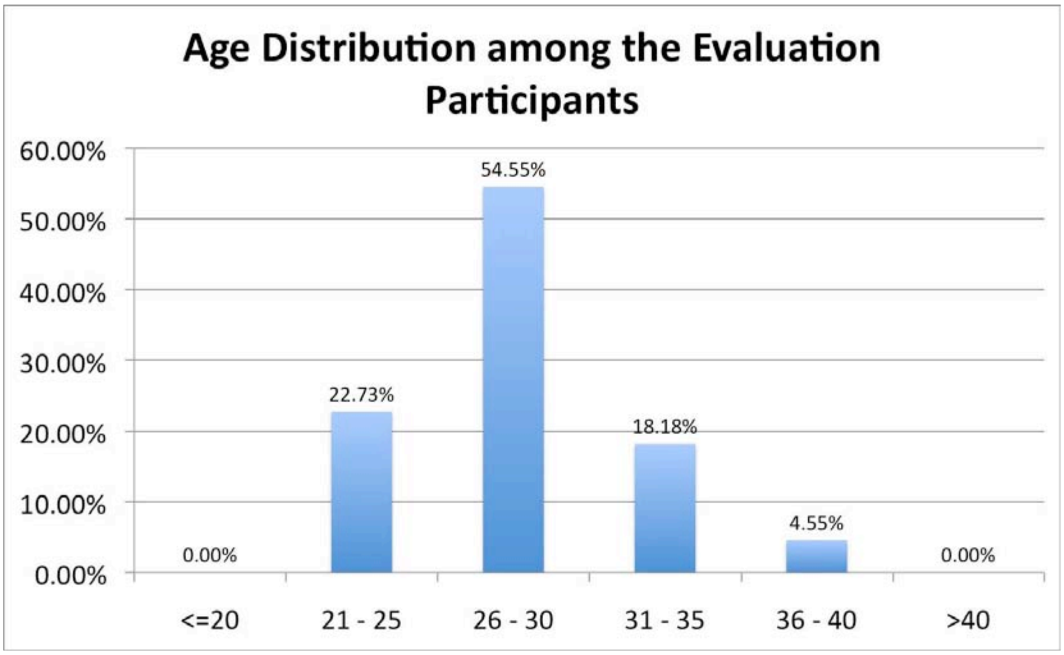


Figure 7.3: Age distribution among the evaluation participants

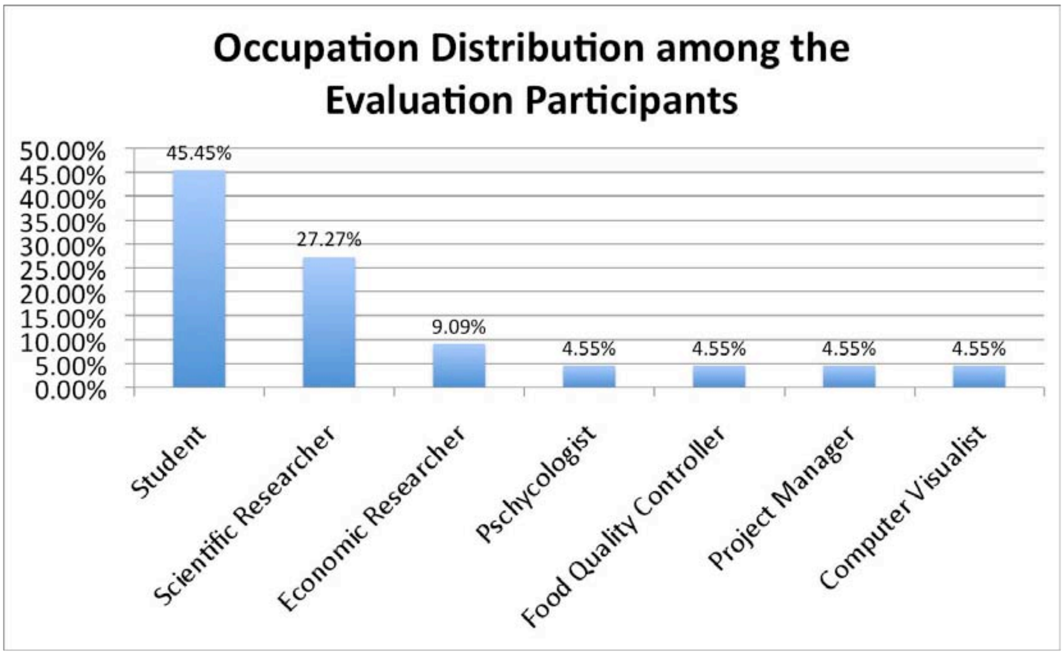


Figure 7.4: Occupation distribution among the evaluation participants

Figure 7.5: Participants during evaluation sessions

they would search for smart objects using UbiLens in three minutes (both in task 1 and task 6).

Searching without UbiLens

Non-tagged environment: without UbiLens

In group one, where the participants did not have any additional aid (i.e. tags), things were a bit tricky. Most of them went for *technical* devices, such as the coffee machine and the TV. Only one have guessed the paper lamp to be smart just because they saw it during our Hydragizer demonstration (cf. see section 6.7—“A Spin-off: Hydragizer”). In average, each participant managed to find two smart objects and made at least one wrong guess before giving up just after one minute of guessing. It was nearly 30% success rate with 36 seconds for each smart object.

Tagged environment: without UbiLens

The participants in group two managed to find five smart objects each with more or less one wrong guess in just over a minute. This makes 80% success rate and 14 seconds to find one smart object. Since each smart object was tagged with QR codes, most of the participants guessed that they should be smart even though the participants looked a bit sceptical about the idea of having a smart painting. Similar to group one, after they were finished finding all the tagged objects, most of them went for technical devices even though no tags were found.

Searching with UbiLens

Trying out both environments

Participants in both groups were given the opportunity to try out searching for smart objects in both non-tagged and tagged environments. Each participant were given three minutes. However, several participants managed to find all smart objects in less than three.

Non-tagged environment: with UbiLens

The participants spent almost three minutes to find smart objects in the non-tagged environment. Compared to the searching without UbiLens, no participant gave up searching. In average, almost five smart objects were found in three minutes window. This makes 68.83% of success rate, more than a double of the success rate without UbiLens. 35 seconds were spent to find a single smart object, which is not a big improvement from without using UbiLens.

Tagged environment: with UbiLens

In the tagged environment, the participants performed considerably worse than without using UbiLens. They needed 23 seconds to find one smart object; 9 seconds slower than without UbiLens. However, the success rate increased to 84.09%. In this environment, most participant tried to recognise the smart object by pointing the camera at the QR code (cf. Figure 7.6) instead of pointing at the whole object.

Hidden Objects

Focus on searching

Both groups were asked whether they noticed the different visual cues displayed by UbiLens whenever it found visible or hidden objects. Only 6 out of 22 participants notice the hidden cues. The others either did not notice or did not find the smart object at all. When they were asked why they did not notice. They said that it is not because the cues were not intuitive but they focussed on searching for smart objects rather than reading the text or paying attention to what is displayed on the screen.

Discussion

The participants enjoyed UbiLens

Even though during the evaluation UbiLens did not increase the speed of the searching activity, the participants were enjoying the sessions with UbiLens. It can be seen from the time they spent searching for smart objects. Without UbiLens, they were clueless and gave up easily, especially in the non-tagged environment. Some participants even said that they wished they had more time using UbiLens. They said that searching for smart objects was similar to searching for eggs during Easter.

Comparing results

Several problems, especially the ones caused by Object Recognition Component (cf. section 7.6—"Problems Encountered"), made up the low performance of UbiLens. Table 7.3—"Participants performance in searching for smart objects" compares the participants performance in finding smart objects with and without UbiLens. Nevertheless, these numbers proved that tags did increase the speed in finding smart objects because the participants were sure that they were smart before testing it with UbiLens.

Measurement	**Non-tagged**		**Tagged**	
	Average	Variance	Average	Variance
Found smart objects				
- without UbiLens	2 out of 7	0.19	5 out of 6	1.42
- with UbiLens	5 out of 7	1.88	5 out of 6	1.32
Success rate				
- without UbiLens	28.57 %	0.00	80.30 %	0.04
- with UbiLens	68.83 %	0.04	84.09 %	0.04
Time to find smart objects				
- without UbiLens	1.23 mins	0.21	1.27 mins	0.28
- with UbiLens	2.81 mins	0.14	1.94 mins	0.82
Time needed to find one smart object				
- without UbiLens	0.61 mins	0.21	0.26 mins	0.28
- with UbiLens	0.58 mins	0.14	0.38 mins	0.82

Table 7.3: Participants performance in searching for smart objects

Task 2: Checking Out the Services Offered

Consuming services

UbiLens is not only about finding smart objects. After playing around finding smart objects in the room, both groups were asked to consume the services that the smart objects offered. Since there was no difference in consuming in either non-tagged or tagged environment, the results were combined and calculated.

A service to buy

The participants were asked to recognise either the smart

Figure 7.6: A participant is pointing at the QR code

book or the smart painting using UbiLens. One of the services offered by both objects is to buy them online. What the participants had to do was to recognise the object, tap the corresponding icon, and choose the "Buy now" menu. In average, this action took 42 seconds.

A service to make coffee

Similar to the steps above, but this time the participants were asked to make coffee from a coffee machine, which was simulated by a laptop running a Video Player Service that plays a video of a coffee machine making coffee. Most of the participants found it hilarious. In average, 55 seconds were spent to do this sub-task.

Discussion

What took so long?

Again the long time needed to perform such a simple activity was due to the performance of the Object Recognition Component (see previous task's discussion section). Apart from that, several participants tried to do something else in the middle of the task. For example, instead of buying the painting, the participants checked the painter information. This makes it difficult to measure the real time needed to do the task.

Task 3: Take Away Some Smart Objects

How to bookmark

In this task, the participants were introduced with the concept of bookmarking smart objects. Bookmarked smart objects could be accessed elsewhere. There are two ways how smart objects can be bookmarked: by tapping on the bookmark button and by dragging the smart object's icon to the bookmark drawer.

Bookmarking using bookmark icon

At first, the participants were asked to open the smart object's menu. By looking at the menu, the participants should be able to bookmark the smart object. This bookmark technique took approximately 14 seconds in average. Several participants said that the bookmark icon is too small and unrecognisable. One participant was confused with the concept of bookmarking but did not ask further. Two participants, familiar with bookmarking webpages, tried to open a service, such as "Get info" and bookmarked the webpage.

Bookmarking by dragging smart object's icon

Another way to bookmark is to drag the smart object's icon to the bookmark icon or bookmark drawer located on the left side of the screen. This technique took 17 seconds in average. The participants were mostly having difficulty in touching the smart object's icon which kept jumping around (cf. section 7.6.2—"Usability Problems").

Discussion

Which bookmark technique to use?

Based on the user experience during the evaluation, creating bookmark by dragging took a lot more time than creating bookmark by tapping the bookmark button. However, this was not really accurate. What happened was that the time to bookmark using bookmark button was measured after the smart object's menu has already opened. Therefore, the time mentioned here is the time needed to search for the bookmark button. With the bookmarking by dragging, the time started before the smart object has been recognised by the Object Recognition Component. Thus, it can be said that whenever the smart object's menu has been opened the bookmark button technique is better than dragging. On the other hand, when the menu has not been opened yet, the dragging would be faster.

Task 4: Opening Bookmarks & Task 6: Drag&Drop

Task 4 was divided into two sub-tasks. The first sub-task was to open a bookmark on the phone while the second one was to open a bookmark on another smart object, the so-called Drag&Drop. In task 6, the Drag&Drop was revisited.

Tasks to do

Opening bookmark drawer was not a problem at all. It took approximately 3 seconds for the participants to realise the bookmark icon on the top left corner of the screen has to be tapped to reveal a bookmark drawer. The bookmark drawer contains all bookmarks stored.

Bookmark drawer

After finding a certain smart object bookmark, the participants were told about the concept of Drag&Drop. The evaluator then asked the participants about how they would do the Drag&Drop. 7 out of 22 participants discovered how to do the Drag&Drop by only reading the description of Drag&Drop. Others needed help. In average, it took 25 seconds to do the drop (i.e. one part of the Drag&Drop). Two participants suggested to bookmark both objects and perform Drag&Drop inside the bookmark drawer.

The Drop

In task 6, the participants had to perform a full Drag&Drop, namely from bookmarking the provider smart object to dropping it to the consumer smart object. They spent 54 seconds in average to do this activity. Four participants bookmarked the provider via the bookmark button while the others performed dragging. Two participants bookmarked both smart objects.

Full Drag&Drop

7.5.2 User Satisfaction Questionnaire

Upon the completion of all the tasks, the participants were required to fill in the user satisfaction questionnaire. This questionnaire was composed by 25 questions. The answers to the questionnaire were based on Likert scale with 1 being strongly disagree to 5 being strongly agree.

SUMI-based Questionnaire

SUMI questionnaire

Among the 25 questions in the questionnaire, 12 questions came from the *Software Usability Measurement Inventory (SUMI)* questionnaire. SUMI questionnaire has a set of 50 questions used to measure the usability of a software product. Among these 50 questions, 12 questions, which suit best with the goal of the evaluation, were chosen.

No significant difference

For each question, the average of the total result, the average of the result from group 1, and the average of the result from group 2 were calculated. Additionally, the variance for each question was also measured. Figure 7.7 depicts the result of the SUMI-based questionnaire. As seen on the figure, both groups do not have significant difference in opinions about the usability of UbiLens.

Unexpected results

The result of the SUMI-based questionnaire reflects the developer's expectation. However, there were two questions of which results were not as expected.

#2 : The system has at some time stopped unexpectedly

Question number 2, which is whether the system has stopped unexpectedly, resulted in "neither agree nor disagree". By looking at the variance, the number is considered high compared to the other questions. It means that several participants did encounter system error while the others did not. Looking back at the evaluation sessions, it is true that for several participants the system did stop unexpectedly. However, the problem was not due to UbiLens itself but rather because of hardware failures. Further details regarding this problem are presented in section 7.6.3—"Miscellaneous Problems".

#8: The speed of this system is fast enough

Question number 8 only gained a "neither agree nor disagree". The speed of the system depends on two entities. One is the speed of the wireless router in forwarding the camera image from the mobile phone to the Object Recognition Component. The other is the speed of the Object Recognition Component performing the recognition process. Mostly the later was responsible for this speed problem. See section 7.6.1—"Technical Problems" for further details.

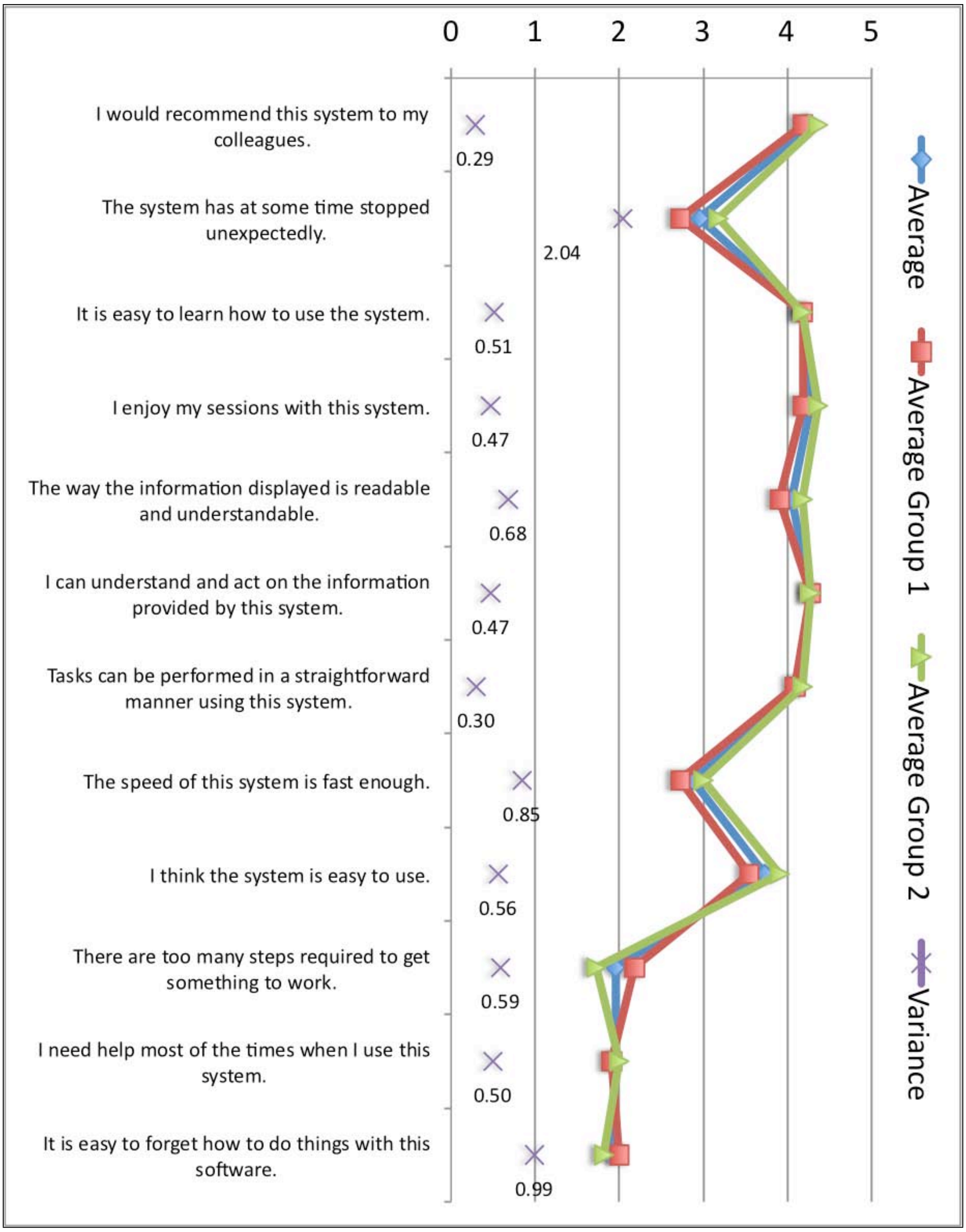


Figure 7.7: SUMI-based user satisfaction questionnaire result using Likert scale with 1 strongly disagree to 5 strongly agree

Custom Questionnaire

Specific questions

The rest of the questionnaire was aimed at getting the participants' opinions about certain concept and usability specific to UbiLens. Similar to the SUMI-based questionnaire, for each questionnaire the average of the whole answers, the answers from group 1, and the answers from group 2 were calculated separately. Figure 7.8 illustrates the result of the questionnaire along with the variance for each question.

UbiLens is a candidate in bridging users, real world objects, and their services

The main goal of this whole thesis project is to prove whether UbiLens is a candidate to bridge the interaction between the users, real world objects, and their services. With the focus on the visualisation of and interaction with real world objects, the outcome of question 3 and 7 answered this goal. In average, the participants agreed that they preferred using UbiLens than guessing to find smart objects (i.e. question 3). Additionally, they also said that UbiLens helped finding hidden information (services) on certain devices (i.e. question 7).

Non-tagged vs. tagged environment

At the beginning of the evaluation, a goal was set to investigate the non-tagged and tagged environments. Question number 2 tells that the participants found the tags spoiled the appearance of the smart objects. However, in question 4, they were not sure whether additional tags are necessary. Both questions have slightly high variance value (i.e. 1.20 and 1.41 respectively), which means that each participant has different opinions about this matter. Until this point, it cannot be concluded whether additional tags are needed by the users. However, based on the participants performance during the evaluation sessions, finding smart objects with additional tags were faster than without.

Visible vs. hidden objects

The visual cues chosen to differentiate visible and hidden smart objects were not optimal. In question 8, the participants stated that they were not really sure whether they could differentiate both objects. However, with the variance value of 1.17, it can be said that whether the individuals could differentiate the visible and hidden objects, depends on the individuals themselves.

Bookmarking

The concept of not being able to bookmark every smart

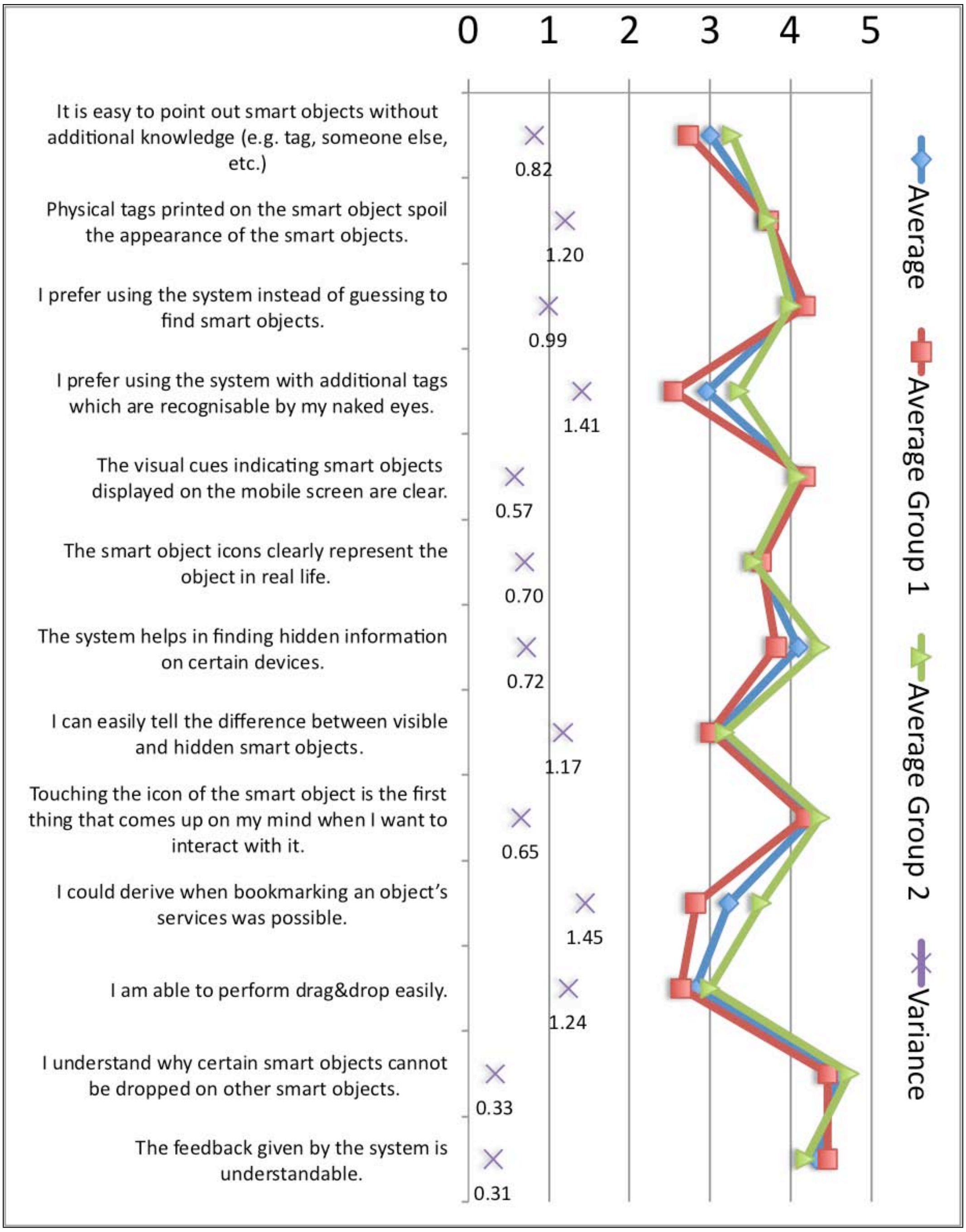


Figure 7.8: User satisfaction questionnaire result using Likert scale with 1 strongly disagree to 5 strongly agree

objects was not widely accepted. This can be seen by the variance of the answers (i.e. 1.45). Even though during the evaluation each participant was explained about this concept, some of them still thought that bookmarking should be possible for every object, just like bookmarking webpages.

Drag&Drop

Many participants were having difficulties in performing Drag&Drop, although some of them performed it easily and quickly. In question 11, they stated that performing Drag&Drop is neither easy nor difficult. With the variance of 1.24, it can be said that this depends on each individual. Not only this, but also several aspects were responsible for this problem. Consult section 7.6—"Problems Encountered" for further details.

7.6 Problems Encountered

Several problems were spotted during the evaluation. These problems were divided into three categories: technical, usability, and miscellaneous problems.

7.6.1 Technical Problems

Object Recognition Component problems

During the evaluation sessions, UbiLens did not perform optimal in finding smart objects in both environments. In ideal case, when users point their mobile phone's camera to the direction of the object, UbiLens gives feedback immediately. However, this immediate feedback depends highly on the performance of the Object Recognition Component (cf. section 6.4—"Object Recognition Component"). The image recognition component used during the evaluation was only capable of recognising one object at a time. The participant even had to point the mobile phone's camera very close to the object so that only the object was visible on the screen. Thus, the performance of the participants in completing the tasks was affected and became slower compared to searching without UbiLens.

Lighting problems

Furthermore, the image recognition component works best when the room is lit with natural light. However, since both environments were set in a room without any windows, several ceiling lamps were used. These artificial lights did not co-operate well with the image recognition component. Therefore, there were considerably long delays for the image recognition component to recognise a certain object.

Unable to recognise QR codes

Most of the participants in the tagged environment tried to recognise the objects by their QR codes, which actually came naturally. They spent a considerable amount of time trying. Even though it seems that by recognising QR codes might result in faster recognition, this was not the case. Somehow the image recognition component performed better in recognising the object and the QR code as a whole rather than the QR code alone.

7.6.2 Usability Problems

Too small icons

Several usability problems were also found during the evaluation. One problem was about the icon of the object being too small. Several participants have big fingers and the icons were just too small for the fingers.

Jumping icons problem

Another problem was the icons were jumping around the screen. The participants were having difficulty to tap on the icons. The location of the icon was determined by the middle point of the image in the image recognition database used to recognise the object. For each smart object, the image recognition component has several reference images. The jumping icons happened because, for instance, at one point the image recognition component recognised the object with image A as a reference while at another time with image B. Both images have different middle point if it is translated to the screen coordinates. Figure 7.9 depicts the jumping icons problem. This created problems when the participants wanted to select or drag a certain object.

On-Off visual cues

Even worse than jumping icons, at some point, the Object Recognition Component did not recognise an object fluidly. When the participants were pointing at the objects, the visual cues went on and off. When they went off it means that the Object Recognition Component did not recognise the object, even though the participants did not move their

Figure 7.9: Jumping icons due to different reference images

camera away. The on and off visual cues made the selection of a smart object and the Drag&Drop difficult to perform.

7.6.3 Miscellaneous Problems

Familiarity with the mobile phone

Not every participant was familiar with touch screen phones, let alone Android phones. The lack of experience slowed down the interaction between the participants and UbiLens. One example is the pressure level. Android phone acknowledges touching when a soft pressure is applied on the phone's screen. If the pressure gets too harsh, the phone ignores it. Several participants, never handling a touch screen phone before, touched the phone screen real hard hoping the phone would response faster.

Hardware failure

Two essential devices failed during several evaluation sessions. This resulted in system stopping unexpectedly. The first device is the laptop used as Object Recognition Component, Object Information Component, as well as Coffee Machine Service. This laptop automatically restarted in average three times a day. There was not much to be done except restarting the components and service every time the laptop restarted. A laptop replacement was not possible because the deployment of the Object Information Component took a considerable amount of time. The second device is the wireless router. The wireless router was responsible for the data transmission between the components of UbiLens. At some point in time, it just stopped working. There were no apparent reasons for this behaviour. Thus, every now and then the router had to be restarted.

7.7 Evaluation Conclusion

A great success

Overall, the evaluation was a great success. With an average of five participants per day, it can be said that UbiLens attracted people's attention to try out. The scenarios presented to the participants not only helped the participants to visualise the flow of UbiLens but also entertained them at the same time so they did not get bored. Even though there were several problems encountered, the result gathered from the user satisfaction questionnaire tells that UbiLens is a possible candidate to bridge the users, real world objects, and their services. A short demo video based on the UbiLens evaluation will be published online[1] in the near future.

[1] http://www.youtube.com/user/MobileKnowledge

Chapter 8

Summary and View of the Future

"The future belongs to those who believe in the beauty of their dreams."

—Eleanor Roosevelt

With the user evaluations done for the UbiLens prototype, this thesis project comes to an end. The first section of this chapter summarises and reflects what has been done throughout this thesis project. Even though the project is ended, it does not mean that no further improvement is possible. The second section presents several possibilities in improving UbiLens.

8.1 Summary and Contributions

The importance of discovery and interaction

With the emergence of disappearing computers and virtual services which could be attached to any ordinary objects (i.e. smart objects), a tool to help people finding these objects, interacting with them, and consuming the attached services is needed. UbiLens tries to answer this call by combining augmented reality, real world interactions, and object recognition on mobile phones.

Three cycle project

Claimed to provide a mediator between users, real world objects, and their services, this thesis project used user centred design and iterative and incremental development methodologies as the development guidelines. The project was divided into three cycles in which users were involved in the development process. In the first cycle, a set of preliminary user interface for UbiLens was designed and a survey was put online to asses the user interface. Once the result of the preliminary user questionnaire was gathered, paper prototype studies completed the second cycle. The third cycle dealt with the implementation of UbiLens on Android phones along with the user evaluation.

System architecture

The UbiLens system consists of five components, namely the Smart Object, the Smart Object Web Service(s), the Object Recognition Component, the Object Information Component, and the UbiLens mobile application. Each component communicates with each other via wireless network connection. The UbiLens mobile application forwards the image captured by the mobile phone's camera to the Object Recognition Component to be recognised. Upon recognition, UbiLens requests the Object Information Component to forward the corresponding object information. Whenever users interact with the services offered by the Smart Object, UbiLens communicates with the Smart Object Web Service via a certain connection protocol specified by the Smart Object Web Service.

Visualisation and Interaction

UbiLens focuses on visualising and bridging the interaction between users, real world objects, and their services. When a smart object is spotted on the mobile phone's camera, UbiLens displays visual cues to tell the users about the discovery. Furthermore, it allows users to interact with the smart objects, such as consuming services, bookmarking, and Drag&Drop.

User evaluation

The result of the evaluation reveals that UbiLens is capable of mediating interaction between users and virtual services attached on physical objects. It is easy to use and simple to learn. But the most important thing is that the participants, as representative of UbiLens users, enjoyed their sessions with UbiLens.

8.1.1 UbiLens Contributions

Although this thesis project did not answer all research questions defined prior to the beginning of the project due to hardware or component limitations, it contributes to the interaction between the real and the virtual worlds. In this section, the research questions will be answered.

Answering questions

At the moment, UbiLens uses image recognition algorithm to recognise smart objects. Users only need to pan the mobile phone's camera and UbiLens tells whether an object is smart. UbiLens allows this recognition component to be exchanged or extended with any recognition algorithms. The change will not lead to any major changes in UbiLens codes.

How can smart objects be recognised?

Upon recognising a smart object, UbiLens displays the corresponding smart object's icon along with a bright border. In ideal case, the border shall surround the smart object. Since this is currently not possible, instead giving border to the smart object, UbiLens draws a frame border along the border of the screen. These visual cues are effective to attract users' attention. This has been proved during the evaluation. Without being explained, the participants knew that these cues meant that the object they were pointing at was smart.

Which visual methods are effective in giving cues that an object has services attached?

Dashed borders along the border of the screen and the smart object's icon were chosen to tell users that an object is hidden. According to the user evaluation, these cues were not optimal. The participants barely noticed the difference between the visible and hidden cues. However, the participants said that they did not really care about whether the object was hidden, they only cared whether they found a smart object and could consume the services offered.

Which visual cues are effective to indicate that the objects are occluded by other objects?

In this thesis project, ambiguity was not investigated further because the recognition component used is only capable of recognising one object at a time. Consult section8.2—"View of the Future" for alternatives.

How can selection ambiguity be solved when two or more smart objects are closed together?

UbiLens was developed on a touch screen mobile phone. Therefore, most interactions use the touch ability of the screen. In someway, this interaction can be considered as

How do users interact with smart objects via a mobile phone?

direct manipulation since the users see the objects on the phone's screen and touch the objects via the phone's screen.

What kind of interactions are possible in smart environments?

When people want to interact with an object in real life, they first will touch it. This concept is used in UbiLens. When users see the smart object on the mobile phone's screen, they only need to touch it to interact with them. The same concept also applies to bookmark by dragging the smart object's icon to the bookmark drawer. In real life, this corresponds with taking away an object and putting it in a pocket for later use. The Drag&Drop technique allows users to ask a smart object to consume another smart object. In this technique, users take the smart object and drop it onto another smart object. An analogy of this activity would be to take pasta from a kitchen drawer and drop it into a pan so that the pan cooks the pasta.

8.2 View of the Future

Time and component limitations

Nothing is perfect. Due to limitations of both time and component capabilities, several features remained left out from this thesis project. Below is several suggestions that could make up the fourth, the fifth, the sixth cycle, and so on.

Occlusion problem

At the beginning of this thesis project, a set of questions were defined to be answered upon completion of this project. One of the questions was about *occlusion*. As mentioned earlier, in every day life it is common to have objects occluded by other objects when they are seen from one view point (i.e. hidden). In this project, UbiLens shows an alternative on how to convey this message to the users. Before the visual cues for hidden objects are displayed, UbiLens needs to know whether the object recognised by the Object Recognition Component is at the moment hidden from the user's view point. It is obvious that the one who should tell UbiLens whether a smart object is hidden is the Object Recognition Component. However, in the currently used Object Recognition Component, namely image recognition algorithm, it is not possible to recognise an object if it is hidden. Partial occlusion may be recognisable but not the full one. This problem can be solved by using other recognition algorithms which are not only able to recognise objects but also tell the location of the object in 3D space. One alternative would be to use indoor localisation system. Another alternative is to combine available recognition algorithms. Whatever recognition algorithms are chosen, exchanging the Object Recognition Component does not lead to any conflicts with the other components.

Ambiguity problem

The image recognition algorithm used in this project also raises another problem, namely the *ambiguity* problem. Ambiguity problem rises when several smart objects are located nearby each other. The image recognition algorithm can only recognise one object per camera image frame. Therefore, no further investigation about ambiguity has been made during the development of this project. An alternative would be to use image recognition algorithm which is capable of recognising more than one objects per camera image frame. In the current implementation, if the recognition component is capable of recognising more than one object, users only need to tap the icon of the smart object they want to interact with. However, this can not be done for Drag&Drop activity. Therefore, other solutions must be investigated.

Out-of-range smart objects

The added feature mentioned during the preliminary user questionnaire that is to have visual cues to indicate out-of-camera-range smart objects were not further investigated during the implementation phase of this thesis project. The reason lies on the image recognition algorithm used. As mentioned before, if an object is not captured by the camera, the image recognition algorithm cannot recognise it. Therefore, other additional recognition algorithms which are able to calculate the location of nearby objects, such as indoor localisation system and wireless tags (e.g. RFID, NFC, Bluetooth), are needed.

Colour blindness

UbiLens plays a lot with colours in giving visual cues. This can be tricky when UbiLens is used by colour blind people. 99% colour blind people have difficulties in determining red and green [Col, 2010]. UbiLens uses red and green to tell whether two objects are compatible with each other during Drag&Drop process. In order to give support to colour blind people, other visual cues need to be designed. These visual cues should, if possible, avoid any colour references. One alternative is to display check mark (✓) for compatible and cross (×) for incompatible. These cues can be either displayed in the middle of the screen or used as border patterns.

Provider service

In Drag&Drop activity, two services from two smart objects are involved. One service acts as the provider service and the other as the consumer service. The provider service is one among the services offered by the provider object. In the current implementation of UbiLens, users choose one smart object from the bookmark list to be the provider object and UbiLens decides which service is the provider service. What happened is that UbiLens iterates through the list of the smart object's offered services and chooses the first service of which type belongs to provider types (i.e. Audio and Video) to be the provider service. Obviously, this decision is not an effective one. If a smart object has two services which both have the service type that belongs to provider types, then the second service will never be chosen by UbiLens. Thus, users need to be given the freedom to choose which service they want to consume. One alternative is by bookmarking a service instead of a smart object. In this way, each bookmark corresponds to exactly one service.

Consumer service

The provider service problem above also happens during the selection of the consumer service. When users drop the provider smart object/service to the consumer service, UbiLens chooses the consumer service based on the order of the service. One alternative to address this problem is to pop up a menu asking users which service they want to use to consume the provider service.

Security and privacy issues

This thesis project does not concern with any security and privacy issues. One issue includes the accessibility of the smart object and its services. Certain smart objects or services should not be consumed by anyone. Take a printer as an example. A printer offers services to print a document and to change the printing fee. Obviously the second service should only be consumed by certain people. A password can be used to secure the consumption of smart objects services. If this is not enough, more advanced security mechanism should be elaborated with UbiLens.

In this thesis project, UbiLens was meant to help users find and consume services scattered in the environment. Apart from that, another project, called *Hydragizer* by Fraunhofer FIT, has been built based on the visualisation technique of UbiLens. It displays the current energy consumption of the smart object. This project was exhibited at CeBIT 2010 in Hannover. Games, such as scavenger hunt or hide and seek, can also be built based on the UbiLens' visualisation technique.

UbiLens' visualisation technique

Overall, the aims to find a direct visualisation and interaction technique were met. Advantages and disadvantages of UbiLens were discussed. Several ideas and open issues for future work were outlined as well. The positive feedback from the evaluation participants opens up a whole new opportunity to extend UbiLens further.

Extending UbiLens

Appendix A

Preliminary User Questionnaire Form

This is the form used to gather preliminary user requirements. The form was published online and distributed to the participants via http://www.questionpro.com.

Preliminary User Questionnaire

General Instruction

The purpose of this questionnaire is to find out how you, as a user, interact with an upcoming mobile application, which will be developed as a Master Thesis project.

This questionnaire has 22 questions and takes approximately 15-20 minutes to complete. Your personal identifying information will not be published. Please answer all questions at your best. Further suggestions are always welcome. Thank you for your time!

Vina Wibowo 1 22 June 2009

Figure A.1: Preliminary user questionnaire form - page 1

Preliminary User Questionnaire

Searching for Smart Objects

Imagine that you are in a living room filled with ordinary-everyday-life objects, such as TV set, sofa, table, cupboard, etc. Among these objects, some of them are considered to be "smart". Currently you are holding a mobile phone along with XApp installed on it. XApp is a mobile application which is able to tell you which objects are "smart" and which are not.

1. How do you search for "smart" objects? Please choose one.

	By panning the camera through the whole room
	By getting closer to a particular object and focusing the phone's camera for a couple of seconds
	By taking picture of a particular object using the phone's camera
	Other:

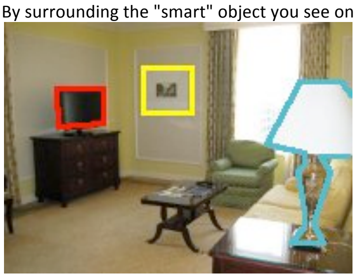
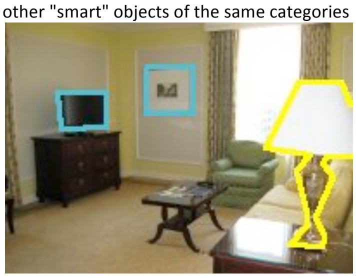
Vina Wibowo 2 22 June 2009

Figure A.2: Preliminary user questionnaire form - page 2

Preliminary User Questionnaire

Visualization of Smart Objects

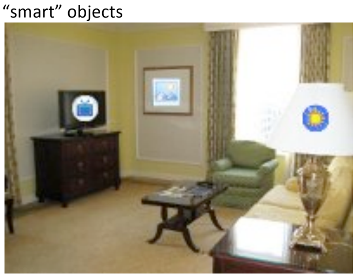
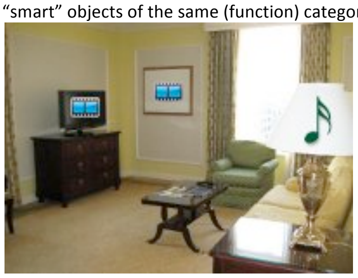
2. How do you want XApp to tell you that an object is “smart”? Please choose one.

	By surrounding the “smart” object you see on the screen with a common colored border 
	By surrounding the "smart" object you see on the screen with a unique colored border 
	By surrounding the "smart" object you see on the screen with a same colored border as other "smart" objects of the same categories 

Vina Wibowo 3 22 June 2009

Figure A.3: Preliminary user questionnaire form - page 3

Preliminary User Questionnaire

	By displaying a particular icon nearby the "smart" object using the same icon for every "smart" objects 
	By displaying a particular icon nearby the "smart" object using a unique icon for each "smart" objects 
	By displaying a particular icon nearby the "smart" object using the same icon for every "smart" objects of the same (function) category 
	Others:

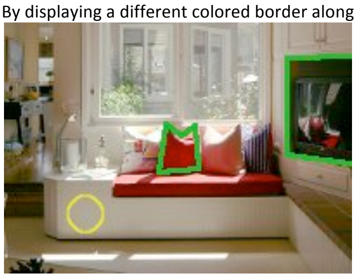
Vina Wibowo 4 22 June 2009

Figure A.4: Preliminary user questionnaire form - page 4

Preliminary User Questionnaire

3. Sometimes other smart objects, other objects, or people occlude a smart object and you cannot see it through the phone's screen.

   How do you expect XApp to tell you that there is a "smart" object behind other objects? Please choose one.

	By displaying a dashed colored border along the "smart" object shape 
	By displaying a different colored border along the "smart" object shape 
	By displaying an icon with different color than the normal icon (used in q. 2) nearby the location of the occluded object 

Vina Wibowo 5 14 June 2009

Figure A.5: Preliminary user questionnaire form - page 5

Preliminary User Questionnaire

	By displaying the icon more transparent than normal (used in q. 2) nearby the location of the occluded object 
	By displaying an icon different from the normal icon (used in q. 2) nearby the location of the occluded object 
	Other:

Vina Wibowo 6 22 June 2009

Figure A.6: Preliminary user questionnaire form - page 6

Preliminary User Questionnaire

4. If the “smart” object is out of camera’s view range, how would you like to see them? Please choose one.

	I do not need to see them at all
	XApp should indicate them using arrows 
	XApp should indicate using a Halo 
	Other:

Vina Wibowo 7 22 June 2009

Figure A.7: Preliminary user questionnaire form - page 7

Preliminary User Questionnaire

Selection of Smart Object

XApp can get information or functions belong to "smart" objects. One example is XApp asks a smart TV set what it can do, the TV set answers that it can play movie, and XApp tells you. At first, you need to choose which "smart" object of which information or functions you want to see.

5. How would you choose a "smart" object, which appears on the screen? Please choose one.

	By touching (the screen) inside the border of the object or touching the icon of the object
	By pointing the camera to get only the object image and pressing the camera button
	Other:

6. What do you expect to happen after choosing a particular "smart" object? Please choose one.

	The information and functions are shown on the screen and the screen image freezes.
	Other:

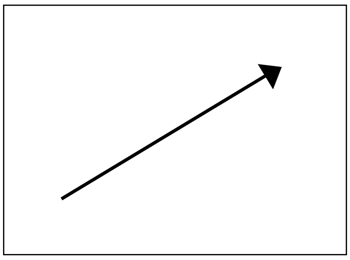
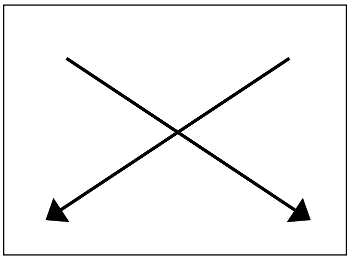
Vina Wibowo 8 22 June 2009

Figure A.8: Preliminary user questionnaire form - page 8

Preliminary User Questionnaire

7. Accidentally, you choose the wrong object.

   What will you do in order to de-select the chosen object and search another object? Please choose one.

	By choosing the provided button (i.e. "Back", "Cancel", "Search Other")
	By doing dust-sweeping gesture using finger on the screen 
	By doing X-figure gesture using finger on the screen 
	Other:

Vina Wibowo 9 22 June 2009

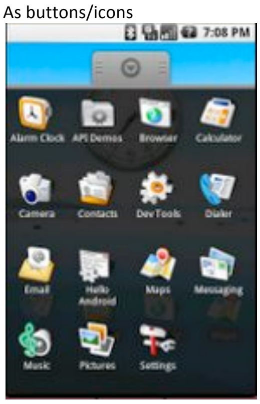
Figure A.9: Preliminary user questionnaire form - page 9

Preliminary User Questionnaire

Smart Object Functions

“Smart” object functions are represented as menus which you can choose whenever you want to interact with it. For example, a “smart” All-In-One printer has “print”, “scan”, and “copy” as functions/menus.

8. How do you like these menus to be presented? Please choose one.

	As list items 
	As buttons/icons 
	As list items or buttons/icons (can be changed later)
	Other:

Vina Wibowo 10 22 June 2009

Figure A.10: Preliminary user questionnaire form - page 10

Preliminary User Questionnaire

9. How will you navigate through the menus? Please choose one.

	By using normal keypad buttons
	By tapping on the screen
	By utilizing tilting gesture (tilting the phone to the front means navigating the menu upward, etc.)
	Other:

Vina Wibowo 11 22 June 2009

Figure A.11: Preliminary user questionnaire form - page 11

Transferring Information between Smart Objects

Two “smart” objects can transfer information to each other through XApp. Imagine that XApp can take a song from a “smart” music CD and play it on a “smart” CD player.

10. If the two objects are seen on the screen together, how would you transfer the information from one object to the other?

	By touching the first object on the screen with finger, dragging the finger to the second object, and releasing the finger
	Other:

11. If two objects are not seen on the screen together, how would you transfer the information from one object to the other?

	By touching the first object on the screen with finger, moving the phone to the direction of the second object so that the finger touches the second object on the screen, then releasing the finger
	By focusing the camera on the first object, pressing the camera button, while still pressing moving the phone so that the camera focuses on the second object, then releasing the camera button
	Other:

Figure A.12: Preliminary user questionnaire form - page 12

Preliminary User Questionnaire

Other Remarks

12. In conclusion, with XApp you can search for "smart" objects and interact with them.

    What requirements do you expect from the application?

13. Please give some suggestions, comments, or remarks which are helpful for the development of this project.

Vina Wibowo 13 22 June 2009

Figure A.13: Preliminary user questionnaire form - page 13

Preliminary User Questionnaire

Personal Information

14. How old are you?

15. What is your gender?

	Male
	Female

16. What is your occupation?

17. Have you ever taken Computer Science studies (and the like)?

	Yes
	No

18. Have you ever developed mobile applications?

	Yes
	No

19. How long have you been using mobile phones?

20. Please specify all mobile phone brands you have used.

	Nokia
	Sony Ericsson
	Samsung
	Siemens
	HTC
	iPhone
	Other:

Vina Wibowo 14 22 June 2009

Figure A.14: Preliminary user questionnaire form - page 14

21. Rank these mobile phone functionalities below based on what you use most!

	Call
	SMS
	MMS
	Games
	Internet
	GPS/Map
	Calendar/Reminder
	Alarm Clock
	Camera
	Other:

Thank you for taking this questionnaire.
Your response is very important to the development of this project.

Figure A.15: Preliminary user questionnaire form - page 15

Appendix B

Requirements Analysis Based on Volere Card

This is a complete listing of requirement analysis done based on Volere requirements analysis card.

Requirement Nr.: 001	**Requirement Type:** Functional	**Use Cases:**
Description: The system shall help users to find service-enabled objects.		
Rationale: Service-enabled objects are mingled with everyday objects. Users need a mechanism to differentiate between these objects.		
Source: System goal		
Fit Criterion: Users are able to find nearby service-enabled objects by using the system in a mixed environment.		
Customer Satisfactory Rating: 5	**Customer Dissatisfaction Rating:** 5	
Dependencies: 003, 004	**Conflicts:**	
History: Created July 2009	**Supporting Materials:** Thesis proposal	

Figure B.1: Requirement #001

Requirement Nr.: 002	**Requirement Type:** Non Functional - Operational	**Use Cases:**
Description: The system shall run on a mobile device.		
Rationale: Service-enabled objects are everywhere. The mobility of the mobile device helps the mobility of the system.		
Source: System goal		
Fit Criterion: The system is installed and running on a mobile device.		
Customer Satisfactory Rating: 5	**Customer Dissatisfaction Rating:** 5	
Dependencies:	**Conflicts:**	
History: Created July 2009	**Supporting Materials:** Thesis proposal	

Figure B.2: Requirement #002

<table>
<tr><td>Requirement Nr.: 003</td><td colspan="2">Requirement Type: Functional</td><td>Use Cases:</td></tr>
<tr><td colspan="4">Description:
The system shall give hints to users whenever it finds visible service-enabled objects.</td></tr>
<tr><td colspan="4">Rationale:
Service-enabled objects are mingled with everyday objects. Users need a mechanism to differentiate between these objects.</td></tr>
<tr><td colspan="4">Source: System objective</td></tr>
<tr><td colspan="4">Fit Criterion:
Visual hints are given to the users to indicate that a particular object is a service-enabled object.</td></tr>
<tr><td colspan="2">Customer Satisfactory Rating: 5</td><td colspan="2">Customer Dissatisfaction Rating: 5</td></tr>
<tr><td colspan="2">Dependencies: 001</td><td colspan="2">Conflicts:</td></tr>
<tr><td colspan="2">History: Created July 2009</td><td colspan="2">Supporting Materials: Thesis proposal</td></tr>
</table>

Figure B.3: Requirement #003

<table>
<tr><td>Requirement Nr.: 004</td><td colspan="2">Requirement Type: Functional</td><td>Use Cases:</td></tr>
<tr><td colspan="4">Description:
The system shall give hints to users whenever it finds hidden service-enabled objects.</td></tr>
<tr><td colspan="4">Rationale:
In the real world, objects might be occluded by another object when they are seen from one point of view.</td></tr>
<tr><td colspan="4">Source: System objective</td></tr>
<tr><td colspan="4">Fit Criterion:
Users can find service-enabled objects even though they are occluded by other objects when they are seen from user’s point of view.</td></tr>
<tr><td colspan="2">Customer Satisfactory Rating: 3</td><td colspan="2">Customer Dissatisfaction Rating: 2</td></tr>
<tr><td colspan="2">Dependencies: 001</td><td colspan="2">Conflicts:</td></tr>
<tr><td colspan="2">History: Created July 2009</td><td colspan="2">Supporting Materials: Thesis proposal</td></tr>
</table>

Figure B.4: Requirement #004

Requirement Nr.: 005	**Requirement Type:** Functional	**Use Cases:**
Description: The system shall give hints to users whenever it finds service-enabled objects outside the users' view range.		
Rationale: Even though a service-enabled object is not visible from the users' view range, it does not mean that it might not be interesting for the users.		
Source: System objective		
Fit Criterion: The users know that there are other nearby service-enabled objects beyond their view range.		
Customer Satisfactory Rating: 3	**Customer Dissatisfaction Rating:** 1	
Dependencies:	**Conflicts:**	
History: Created July 2009	**Supporting Materials:** Thesis proposal	

Figure B.5: Requirement #005

Requirement Nr.: 006	**Requirement Type:** Functional	**Use Cases:**
Description: The system shall provide mechanisms for interaction between users and service-enabled objects.		
Rationale: After finding out what service-enabled objects are available, the ability to interact with them is the next thing desired.		
Source: System objective		
Fit Criterion: The users are able to consume the services attached to the objects.		
Customer Satisfactory Rating: 5	**Customer Dissatisfaction Rating:** 5	
Dependencies: 013	**Conflicts:**	
History: Created July 2009	**Supporting Materials:** Thesis proposal	

Figure B.6: Requirement #006

Requirement Nr.: 007	**Requirement Type:** Functional	**Use Cases:**
Description: The system shall facilitate information exchanges between service-enabled objects.		
Rationale: The service-enabled objects need to communicate with each other in order to provide service to their consumer.		
Source: System objective		
Fit Criterion: The service-enabled objects are able to exchange information through the system.		
Customer Satisfactory Rating: 4	**Customer Dissatisfaction Rating:** 4	
Dependencies:	**Conflicts:**	
History: Created July 2009	**Supporting Materials:** Thesis proposal	

Figure B.7: Requirement #007

Requirement Nr.: 008	**Requirement Type:** Functional	**Use Cases:**
Description: The system shall help the user in deciding which service-enabled object to use in a certain situation.		
Rationale: With the wide range of services offered by service-enabled objects, users need to decide which service is most suitable to fulfil their needs.		
Source: Preliminary user questionnaire participant		
Fit Criterion: The system introduces rating for each service offered by the objects.		
Customer Satisfactory Rating: 3	**Customer Dissatisfaction Rating:** 1	
Dependencies:	**Conflicts:**	
History: Created July 2009	**Supporting Materials:** Preliminary user questionnaire	

Figure B.8: Requirement #008

<table>
<tr><td>Requirement Nr.: 009</td><td>Requirement Type: Non Functional - Security</td><td>Use Cases:</td></tr>
<tr><td colspan="3">Description:
The system shall handle the access of certain service-enabled objects.</td></tr>
<tr><td colspan="3">Rationale:
Not everyone can interact with a protected service-enabled object.</td></tr>
<tr><td colspan="3">Source: Preliminary user questionnaire participant</td></tr>
<tr><td colspan="3">Fit Criterion:
A security mechanism is introduced to protect the service-enabled objects from unwanted users.</td></tr>
<tr><td>Customer Satisfactory Rating: 2</td><td colspan="2">Customer Dissatisfaction Rating: 1</td></tr>
<tr><td>Dependencies:</td><td colspan="2">Conflicts:</td></tr>
<tr><td>History: Created July 2009</td><td colspan="2">Supporting Materials: Preliminary user questionnaire</td></tr>
</table>

Figure B.9: Requirement #009

<table>
<tr><td>Requirement Nr.: 010</td><td>Requirement Type: Functional</td><td>Use Cases:</td></tr>
<tr><td colspan="3">Description:
The system shall provide mechanism to consume services from service-enabled objects anytime anywhere.</td></tr>
<tr><td colspan="3">Rationale:
Several services are consumed frequently. Users do not want to spend time searching for these services whenever they want to consume them.</td></tr>
<tr><td colspan="3">Source: Preliminary user questionnaire participant</td></tr>
<tr><td colspan="3">Fit Criterion:
A mechanism is introduced to store references to the services.</td></tr>
<tr><td>Customer Satisfactory Rating: 5</td><td colspan="2">Customer Dissatisfaction Rating: 4</td></tr>
<tr><td>Dependencies: 011</td><td colspan="2">Conflicts:</td></tr>
<tr><td>History: Created July 2009</td><td colspan="2">Supporting Materials: Preliminary user questionnaire</td></tr>
</table>

Figure B.10: Requirement #010

Requirement Nr.: 011	Requirement Type: Non Functional - Performance	Use Cases:
Description: The system shall remember the previously seen objects.		
Rationale: The users do not want to spend time searching for the previously seen objects.		
Source: Preliminary user questionnaire participant		
Fit Criterion: A mechanism is introduced to store references to the services.		
Customer Satisfactory Rating: 4	Customer Dissatisfaction Rating: 3	
Dependencies: 010	Conflicts:	
History: Created July 2009	Supporting Materials: Preliminary user questionnaire	

Figure B.11: Requirement #011

Requirement Nr.: 012	Requirement Type: Non Functional – Look and Feel	Use Cases:
Description: The system shall have a minimalist/simple user interface.		
Rationale: Complex user interface tends to confuse users.		
Source: Preliminary user questionnaire participant		
Fit Criterion: The system has user interface with shallow hierarchy.		
Customer Satisfactory Rating: 5	Customer Dissatisfaction Rating: 5	
Dependencies:	Conflicts:	
History: Created July 2009	Supporting Materials: Preliminary user questionnaire	

Figure B.12: Requirement #012

Requirement Nr.: 013	**Requirement Type:** Functional	**Use Cases:**
Description: The system shall be able to act as a remote controller to the service-enabled objects.		
Rationale: Users do not see that they do not actually control the object. Instead, they compare the system as a remote controller controlling a TV set, for example.		
Source: Preliminary user questionnaire participant		
Fit Criterion: The system is able to consume the services attached to the object from distance.		
Customer Satisfactory Rating: 5	**Customer Dissatisfaction Rating:** 5	
Dependencies: 006	**Conflicts:**	
History: Created July 2009	**Supporting Materials:** Preliminary user questionnaire	

Figure B.13: Requirement #013

Requirement Nr.: 014	**Requirement Type:** Non Functional - Usability	**Use Cases:**
Description: The system shall introduce different settings for novice and expert users.		
Rationale: Novice and expert users behave differently in interacting with a system.		
Source: Preliminary user questionnaire participant		
Fit Criterion: The system gives different user interfaces for both novice and expert users.		
Customer Satisfactory Rating: 4	**Customer Dissatisfaction Rating:** 2	
Dependencies: 015	**Conflicts:**	
History: Created July 2009	**Supporting Materials:** Preliminary user questionnaire	

Figure B.14: Requirement #014

Requirement Nr.: 015	**Requirement Type:** Non Functional – Usability	**Use Cases:**
Description: The system shall provide supports for disable people.		
Rationale: Disable people cannot interact with the system as freely as normal people can.		
Source: Preliminary user questionnaire participant		
Fit Criterion: Supports for disable people are introduced.		
Customer Satisfactory Rating: 3	**Customer Dissatisfaction Rating:** 1	
Dependencies: 014	**Conflicts:**	
History: Created July 2009	**Supporting Materials:** Preliminary user questionnaire	

Figure B.15: Requirement #015

Requirement Nr.: 016	**Requirement Type:** Non Functional – Portability	**Use Cases:**
Description: The system shall be device-independent.		
Rationale: With the emergence of mobile devices and their operating systems, the system should be easy to adapt with each device.		
Source: Preliminary user questionnaire participant		
Fit Criterion: The system can run on different mobile devices platform.		
Customer Satisfactory Rating: 5	**Customer Dissatisfaction Rating:** 4	
Dependencies:	**Conflicts:**	
History: Created July 2009	**Supporting Materials:** Preliminary user questionnaire	

Figure B.16: Requirement #016

Requirement Nr.: 017	**Requirement Type:** Non Functional - Performance	**Use Cases:**
Description: The system shall response fast.		
Rationale: Searching for service-enabled objects take considerable amount of time. A responsive system should decrease the time needed.		
Source: Preliminary user questionnaire participant		
Fit Criterion: The system give hints of found objects after a maximum three seconds after it recognise the objects.		
Customer Satisfactory Rating: 5	**Customer Dissatisfaction Rating:** 5	
Dependencies:	**Conflicts:**	
History: Created July 2009	**Supporting Materials:** Preliminary user questionnaire	

Figure B.17: Requirement #017

Requirement Nr.: 018	**Requirement Type:** Non Functional - Usability	**Use Cases:**
Description: The system shall be easy to use by every mobile device users.		
Rationale: Different mobile devices lead to different experience in interacting with the system.		
Source: Preliminary user questionnaire participant		
Fit Criterion: The system shall adopt a general user interface to satisfy every user.		
Customer Satisfactory Rating: 5	**Customer Dissatisfaction Rating:** 5	
Dependencies:	**Conflicts:**	
History: Created July 2009	**Supporting Materials:** Preliminary user questionnaire	

Figure B.18: Requirement #018

Appendix C

Paper Prototype Evaluation Form

This is the form used for paper prototype evaluations. The form was given to the participants prior the beginning of each evaluation session.

Paper Prototype Evaluation Form

Introduction

XApp is a mobile application, which recognizes and communicates with smart objects. A smart object is an ordinary physical object, which has been tagged with information, such as services, some data or files.

Goal

This paper prototype evaluation is designed to evaluate the first prototype of XApp user interface and interaction methods. The evaluation focuses on the naturalness of the interaction, the visual cues given, and the user interface in general.

Method

In this evaluation session, you are equipped with a model of Android phone. Using this phone, you are expected to follow the tasks described below by exploring the XApp by yourself. While doing the tasks, it is required that you say aloud what you are currently thinking (i.e. about the interface, interaction, the meaning of the visual cues, etc.). Each task is followed by several questions and you are expected to answer them. At the end of the evaluation, a short interview will be held to discuss your answers.

There would be an evaluator, who acts as the "computer" of your phone and only answers to your question when it is necessary. A few pictures and audio recordings will be taken. Further comments are always welcome.

Vina Wibowo -1- 10 July 2009

Figure C.1: Paper prototype evaluation form - page 1

Paper Prototype Evaluation Form

About Android

Before you begin with the evaluation, here is a short description about the Android phone buttons and their functions, which might help you in completing this evaluation.

Figure 1: Android Model Phone

Android Phone has six buttons on the front side of the phone [Figure 1]. These buttons include:

1. Menu button : to view a set of menus for the current screen
2. Call button : to call
3. Home button : to go back to the desktop (phone's main screen)
4. Trackball : to navigate through menus, icons, etc.
5. Back button : to go back to one previous screen
6. Hang up button : to hang up call

Vina Wibowo -2- 10 July 2009

Figure C.2: Paper prototype evaluation form - page 2

Paper Prototype Evaluation Form

User Profile

Before you begin with the evaluation, please take some time to answer the following questions.

How old are you?

What is your gender?

	Male
	Female

What is your occupation? Please give a complete description (e.g. Media Informatics master student).

Do you have IT background?

	Yes
	No

Have you ever developed mobile applications?

	Yes
	No

How many years have you been using mobile phone?

Vina Wibowo -3- 10 July 2009

Figure C.3: Paper prototype evaluation form - page 3

Paper Prototype Evaluation Form

Task 1 – Searching for and selection of object

Today you are going to give a presentation to your client at your client's office. Once you got there, an employee brings you to the presentation room. He says that the office has tagged several devices in the presentation room so that they would be recognizable by XApp.

In the presentation room, you need a device, which can help you to display your presentation. Search for a projector and test your presentation in that room.

Which visual cue to visualize the smart object do you think fits best? a. border b. icon c. combination of both
Please give reasons if possible!
How helpful is the arrow in giving visual cue that there are smart objects located beyond the screen?
What do you think of the smart object menu?
Further comments:
Time needed (filled by evaluator):

Vina Wibowo -4- 10 July 2009

Figure C.4: Paper prototype evaluation form - page 4

Paper Prototype Evaluation Form

Task 2 – De-selection of smart object

Since XApp is new to you and you need to wait for one hour before starting the presentation, you decide to explore the room.

Go back to start scanning again by trying these options:

- Beginner : touching the "Scan" icon
- Expert : shaking the phone

Do you think as a beginner you will notice that the icon is used to "scan again"? Or do you need a special button with "Scan" as text?
Further comments:
Time needed (filled by evaluator):

Vina Wibowo -5- 10 July 2009

Figure C.5: Paper prototype evaluation form - page 5

Paper Prototype Evaluation Form

Task 3 – Hidden Objects

Search for a hidden object and the information belong to that object.

Which visual cue to visualize hidden smart objects do you think fits best? a. the dashed border b. the icon with additional image c. combination of both
Please give reasons if possible!
Further comments:
Time needed (filled by evaluator):

Vina Wibowo -6- 10 July 2009

Figure C.6: Paper prototype evaluation form - page 6

Paper Prototype Evaluation Form

Task 4a – Drag&Drop

You spotted before that the portrait hanging on the left side of the projector is also a smart object. You want to set it as the "welcome background image" on the projector canvas you use for the presentation.

Transfer the image file from the portrait to the projector screen by doing the following steps.

1. Focus the camera to the portrait.
2. Touch the portrait for three seconds.
3. Focus the camera to the projector canvas.
4. Touch the projector canvas.

What do you think about these steps? Are they convenience to be done?
What do you think about the visual feedbacks?
Further comments:
Time needed (filled by evaluator):

Vina Wibowo -7- 10 July 2009

Figure C.7: Paper prototype evaluation form - page 7

Paper Prototype Evaluation Form

Task 4b – Quick Drag&Drop

Try to transfer the image file from the portrait to the cupboard by doing the following steps.

1. Focus the camera to the portrait.
2. Drag the portrait to the clipboard (located at the bottom left corner of the screen).
3. Focus the camera to the cupboard.
4. Drag the portrait icon on the clipboard (located at the bottom left corner of the screen) to the cupboard.

What do you think about these steps? Are they convenience to be done?
What do you think about the visual feedbacks?
Further comments:
Time needed (filled by evaluator):

Vina Wibowo -8- 10 July 2009

Figure C.8: Paper prototype evaluation form - page 8

Paper Prototype Evaluation Form

Task 5 – Create bookmark

Find out what devices are inside the cupboard. Save this to bookmark for later use.

Try the following options.

- Beginner : touch the bookmark icon and follow the instruction.
- Expert : drag the cupboard icon to the bookmark icon.

Further comments:
Time needed (filled by evaluator):

Vina Wibowo -9- 10 July 2009

Figure C.9: Paper prototype evaluation form - page 9

Paper Prototype Evaluation Form

Task 6a – Open a bookmark on the phone

Open “movie.avi” bookmark on the phone.

Comments:
Time needed (filled by evaluator):

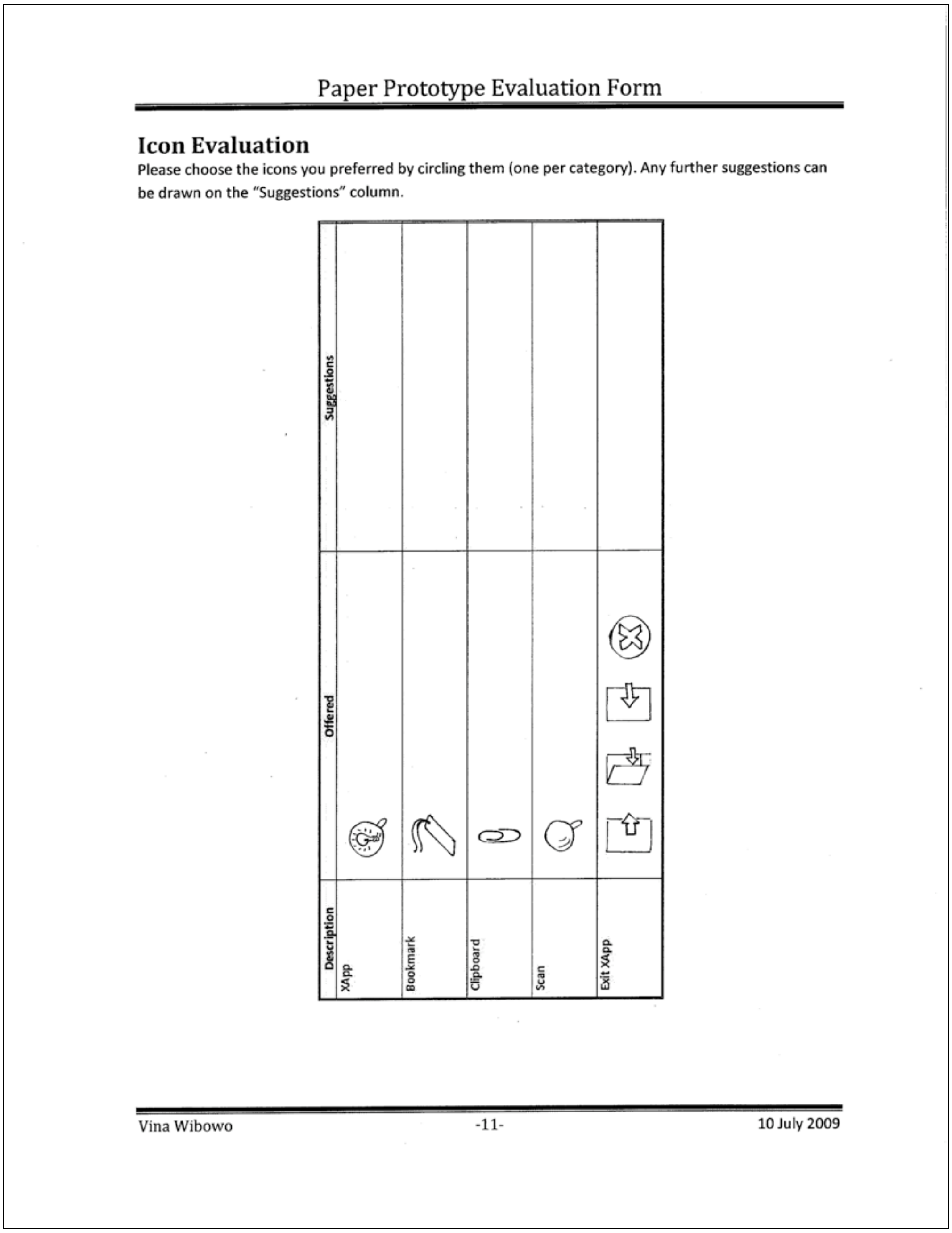
Task 6b – Open bookmark on other smart object

Open bookmark “movie.avi” on the projector.

Comments:
Time needed (filled by evaluator):

Vina Wibowo -10- 10 July 2009

Figure C.10: Paper prototype evaluation form - page 10



Paper Prototype Evaluation Form

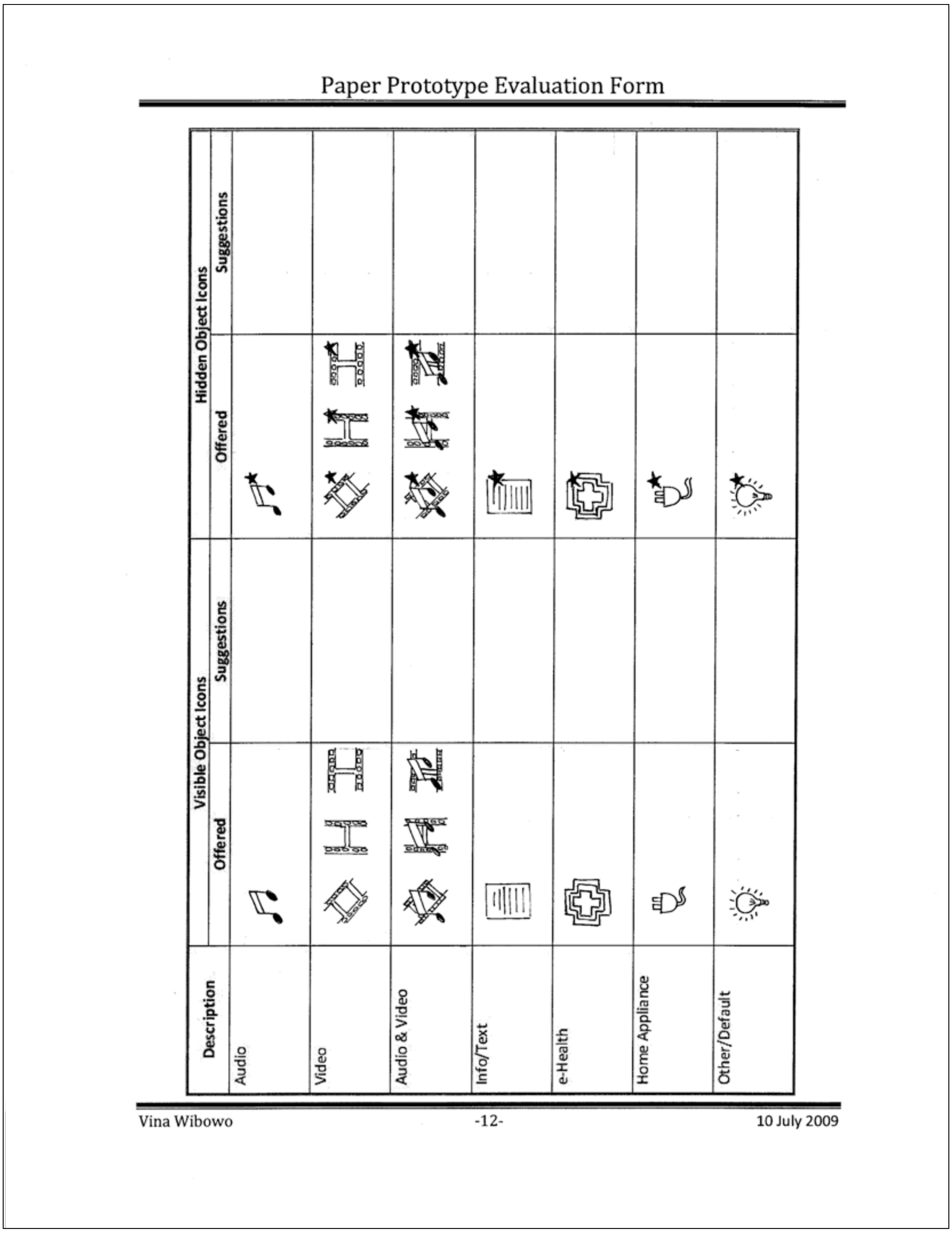
Icon Evaluation

Please choose the icons you preferred by circling them (one per category). Any further suggestions can be drawn on the "Suggestions" column.

Description	Offered	Suggestions
XApp		
Bookmark		
Clipboard		
Scan		
Exit XApp		

Vina Wibowo -11- 10 July 2009

Figure C.11: Paper prototype evaluation form - page 11



Paper Prototype Evaluation Form

Description	Visible Object Icons		Hidden Object Icons	
	Offered	Suggestions	Offered	Suggestions
Audio				
Video				
Audio & Video				
Info/Text				
e-Health				
Home Appliance				
Other/Default				

Vina Wibowo -12- 10 July 2009

Figure C.12: Paper prototype evaluation form - page 12

Paper Prototype Evaluation Form

Placement of Icons

What do you think of the placement of the icons (bookmark, clipboard, scan, and exit)?

Please draw if you have other suggestion on the placement of the icons!

Further Suggestions

Please give further suggestions on how to improve the interaction, the visual cues, and the user interface in general.

Thank you for your time and suggestions.

Vina Wibowo -13- 10 July 2009

Figure C.13: Paper prototype evaluation form - page 13

Paper Prototype Evaluation Form

Interview

(filled by evaluator)

Vina Wibowo -14- 10 July 2009

Figure C.14: Paper prototype evaluation form - page 14

Appendix D

User Evaluation Tasks

This is the list of tasks that were given to the participants during user evaluation sessions. Each of these tasks was written on a piece of index card.

UbiLens User Evaluation Tasks

User Group 1

Non-tagged Environment

Congratulations! You just won a new fully furnished high-tech apartment. You are entering your new apartment for the first time. The representative from the lottery company is coming to show you what you've got.

The representative brings you to the living room, the heart of the apartment. She tells you that several things in the living room are smart. Smart objects offer services beyond their original functions.

Task 1: Searching for smart objects

Can you find all smart objects by just looking at them? Tell the representative the things that you think are smart.

It's almost impossible, isn't it? Ask the representative; it seems that she has something special for you.

The representative will now give you a mobile phone with UbiLens installed on it. Now with UbiLens, try to find smart objects as many as you can. You have three minutes. GO!

Did you notice that one object has a name that does not match with the look of it? Maybe not only the name but also the "smart object sign" is different. Why is it? Share your thoughts with the representative.

Task 2: Checking out the services offered

Once you're playing around finding all the smart objects in the room, you are curious to know what's so smart about smart objects.

Try to interact with that book on the table, for example, using UbiLens. How would you do that? It would be great if you could buy one for your friend's upcoming birthday.

Hey, the coffee machine over there is also smart. Make some coffee out of it.

Task 3: Take away some smart objects

A big poster is hanging on the wall. Apparently it's a project that has something to do with the smartness of your new apartment. And guess what? It's a smart object too. Check what services it offers.

It would be nice if you could bookmark the services offered by the poster so that you can show off to your mates later on about your new high-tech apartment. Go ahead and do that.

Ask the representative if there is anything else you can do to bookmark smart object.

Vina Wibowo -1- 16 April 2010

Figure D.1: User evaluation tasks - page 1

UbiLens User Evaluation Tasks

This activity is called “dragging”. Do this to the paper lamp over there.

Do the same thing to the coffee machine.

Is there something wrong? Share your thoughts.

Task 4: Opening bookmarks

Previously you have bookmarked several smart objects to be used elsewhere (not in front of the object). Find in the UbiLens where you can find all your bookmarks.

Found it? Good. Then open the poster bookmark. Find out more information about the project.

The project has a video to demonstrate what it does. Go watch it.

The mobile phone’s screen is not a good place to watch video. Is there anywhere around you where you can watch video more conveniently?

UbiLens lets you to drop a certain smart object to another smart object. You can use this feature to watch the video on a more appropriate location. Any idea how to do it? The representative will always help you when you need it.

Tagged Environment

Task 5: Searching for smart objects in another environment

Let’s go to the other side of the room.

This side of the room has smart objects too but it looks different than before. Do you notice the difference?

With UbiLens, find as many smart objects as you can in 3 minutes. GO!

Task 6: Drag&Drop

Now find a CD cover and play the music on the HiFi with the drag&drop function you learnt before.

Try dropping the CD on the painting.

With that, you have discovered that you have the coolest apartment in town. It’s time to go to your current place, pack things up, and move to your new home.

Vina Wibowo -2- 16 April 2010

Figure D.2: User evaluation tasks - page 2

UbiLens User Evaluation Tasks

User Group 2

Tagged Environment

Congratulations! You just won a new fully furnished high-tech apartment. You are entering your new apartment for the first time. The representative from the lottery company is coming to show you what you've got.

The representative brings you to the living room, the heart of the apartment. She tells you that several things in the living room are smart. Smart objects offer services beyond their original functions.

Task 1: Searching for smart objects

Can you find all smart objects by just looking at them? Tell the representative the things that you think are smart.

Is there another way? Ask the representative; it seems that she has something special for you.

The representative will now give you a mobile phone with UbiLens installed on it. Now with UbiLens, try to find smart objects as many as you can. You have three minutes. GO!

Did you notice that one object has a name that does not match with the look of it? Maybe not only the name but also the "smart object sign" is different. Why is it? Share your thoughts with the representative.

Task 2: Checking out the services offered

Once you're playing around finding all the smart objects in the room, you are curious to know what's so smart about smart objects.

Try to interact with that painting on the wall, for example, using UbiLens. How would you do that? It would be great if you could buy one for your friend's upcoming birthday.

Hey, the coffee machine over there is also smart. Make some coffee out of it.

Task 3: Take away some smart objects

A music CD is located on the table. You heard that it's from a famous band but you never listen to them yet. Check whether the CD offers some services.

It would be nice if you could bookmark the services offered by the CD so that you can buy the CD later. Go ahead and do that.

Ask the representative if there is anything else you can do to bookmark smart object.

This activity is called "dragging". Do this to the PS3 over there.

Do the same thing to the coffee machine.

Is there something wrong? Share your thoughts.

Vina Wibowo -3- 16 April 2010

Figure D.3: User evaluation tasks - page 3

Task 4: Opening bookmarks

Previously you have bookmarked several smart objects to be used elsewhere (not in front of the object). Find in the UbiLens where you can find all your bookmarks.

Found it? Good. Then open the CD bookmark. Find out more information about the band.

The music CD has sample music. Go listen to it.

The mobile phone is not a good place to listen to music. Is there anywhere around you where you can listen to the music better?

UbiLens lets you to drop a certain smart object to another smart object. You can use this feature to listen to the music on a more appropriate location. Any idea how to do it? The representative will always help you when you need it.

Non-tagged Environment

Task 5: Searching for smart objects in another environment

Let's go to the other side of the room.

This side of the room has smart objects too but it looks different than before. Do you notice the difference?

With UbiLens, find as many smart objects as you can in 3 minutes. GO!

Task 6: Drag&Drop

A big poster is hanging on the wall. Apparently it's a project that has something to do with the smartness of your new apartment. And guess what? It's a smart object too. It offers a video of the project. Watch the video on the big display with the drag&drop function you learnt before.

Try dropping the poster on the coffee machine.

With that, you have discovered that you have the coolest apartment in town. It's time to go to your current place, pack things up, and move to your new home.

Figure D.4: User evaluation tasks - page 4

Appendix E

User Evaluation Form

This is the form used for user evaluation sessions. The first part contains the participant's profile. The second part is the user satisfaction questionnaire which the participants need fill in upon the completion of the evaluation to determine whether UbiLens meets users' expectations. The form was given to the participants prior the beginning of each evaluation session.

UbiLens User Evaluation Form

Welcome to the UbiLens User Evaluation. Today we need you to evaluate UbiLens based on your personal experience in using it during the test. Please keep in mind that we don't judge you how you use the application. For us, whether the application satisfies you, is the most important thing. Therefore, don't be afraid to speak up your opinion whether it is good or bad.

The test will approximately take 30 minutes. During the test you will play a role based on a scenario. In the scenario there are several tasks that you need to do. The tasks are organised using index cards. It is required that you read one card at a time and do what the instruction tells you to do. You don't need to answer all questions. There are there to make you think of them before answering them during the questionnaire.

Besides doing the task, it is important that you say aloud what you are currently thinking. This will help us a lot in improving the application. Remember that the test observer is there to help you when you find any difficulties. Pictures and recordings (audio and/or video) will be taken during the test session. However, your confidentiality will still be our highest priority.

At the end of the test, a short survey will be conducted to measure how satisfy you are in using UbiLens.

Vina Wibowo -1- 08 April 2010

Figure E.1: User evaluation form - page 1

UbiLens User Evaluation Form

User Acknowledgement

I am aware and acknowledged the information that I share during this evaluation session will be used for study purpose only without exposing information that can lead to me and will not be used against me.

______________________ , ______________________ 2010
(Place) (Date)

____
(Signature)

____
(Name)

(For evaluation observer only)
User ID:
User Group:

Vina Wibowo -2- 08 April 2010

Figure E.2: User evaluation form - page 2

UbiLens User Evaluation Form

User Profile

Before you begin with the evaluation, please take some time to answer the following questions.

How old are you?

What is your gender?

	Male
	Female

What is your occupation? Please give a complete description (e.g. Media Informatics master student).

Do you have IT background?

	Yes
	No

Have you ever developed mobile applications?

	Yes
	No

How many years have you been using mobile phone?

08 April 2010 -3- Vina Wibowo

Figure E.3: User evaluation form - page 3

UbiLens User Evaluation Form

UbiLens User Satisfaction Questionnaire

The purpose of this questionnaire is to evaluate the usability of the system you tried just now. Please take some time to fill every field accurately.

Nr.	Statement	Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree
1.	I would recommend this system to my colleagues.					
2.	The system has at some time stopped unexpectedly.					
3.	It is easy to learn how to use the system.					
4.	I enjoy my sessions with this system.					
5.	The way the information displayed is readable and understandable.					
6.	I can understand and act on the information provided by this system.					
7.	Tasks can be performed in a straightforward manner using this system.					
8.	The speed of this system is fast enough.					
9.	I think the system is easy to use.					
10.	There are too many steps required to get something to work.					
11.	I need help most of the times when I use this system.					
12.	It is easy to forget how to do things with this software.					
13.	It is easy to point out smart objects without additional knowledge (e.g. tag, someone else, etc.)					
14.	Physical tags printed on the smart object spoil the appearance of the smart objects.					
15.	I prefer using the system instead of guessing to find smart objects.					
16.	I prefer using the system with additional tags which are recognisable by my naked eyes.					
17.	The visual cues indicating smart objects displayed on the mobile screen are clear.					
18.	The smart object icons clearly represent the object in real life.					
19.	The system helps in finding hidden information on certain devices.					
20.	I can easily tell the difference between visible and hidden smart objects.					

Vina Wibowo -4- 08 April 2010

Figure E.4: User evaluation form - page 4

UbiLens User Evaluation Form

21.	Touching the icon of the smart object is the first thing that comes up on my mind when I want to interact with it.					
	If not, what?					
22.	I could derive when bookmarking an object's services was possible.					
23.	I am able to perform drag&drop easily.					
24.	I understand why certain smart objects cannot be dropped on other smart objects.					
25.	The feedback given by the system is understandable.					
26.	Other remarks/suggestions/comments:					

08 April 2010 -5- Vina Wibowo

Figure E.5: User evaluation form - page 5

Bibliography

What is colorblindness and the different types? Online, April 2010. URL http://colorvisiontesting.com/color2.htm.

Amazon Web Services. Kooaba case study interview: Amazon web services. Online, 2009. URL http://aws.amazon.com/solutions/case-studies/kooaba/.

Atlantic Systems Guild Ltd. Volere requirements resources. Online, 2009. URL http://www.volere.co.uk/.

Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, August 1997.

Rafael Ballagas, Michael Rohs, and Jennifer G. Sheridan. Sweep and point and shoot: phonecam-based interactions for large public displays. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1200–1203, New York, NY, USA, 2005. ACM. ISBN 1-59593-002-7. doi: http://doi.acm.org/10.1145/1056808.1056876.

Rafael Ballagas, Jan Borchers, Michael Rohs, and Jennifer G. Sheridan. The smart phone: A ubiquitous input device. *IEEE Pervasive Computing*, 5(1):70, 2006. ISSN 1536-1268. doi: http://dx.doi.org/10.1109/MPRV.2006.18.

Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80, New York, NY, USA, 1993. ACM. ISBN 0-89791-601-8. doi: http://doi.acm.org/10.1145/166117.166126.

Bluetooth SIG, Inc. Bluetooth. Online, 2009. URL `http://www.bluetooth.com/bluetooth/`.

Davide Bolchini, Diego Pulido, and Anthony Faiola. Feature "paper in screen" prototyping: an agile technique to anticipate the mobile experience. *interactions*, 16(4):29–33, 2009. ISSN 1072-5520. doi: http://doi.acm.org/10.1145/1551986.1551992.

Gregor Broll, John Hamard, Massimo Paolucci, Markus Haarländer, Matthias Wagner, Sven Siorpaes, Enrico Rukzio, Albrecht Schmidt, and Kevin Wiesner. Mobile interaction with web services through associated real world objects. In *MobileHCI '07: Proceedings of the 9th international conference on Human computer interaction with mobile devices and services*, pages 319–321, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-862-6. doi: http://doi.acm.org/10.1145/1377999.1378025.

Gregor Broll, Susanne Keck, Paul Holleis, and Andreas Butz. Improving the accessibility of nfc/rfid-based mobile interaction through learnability and guidance. In *MobileHCI '09: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 2009.

Ed Burnette. *Hello, Android: Introducing Google's Mobile Development Platform*. Pragmatic Bookshelf, 2008. ISBN 1934356174, 9781934356173.

Marco de Sá and Luís Carriço. Lessons from early stages design of mobile applications. In *MobileHCI '08: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 127–136, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-952-4. doi: http://doi.acm.org/10.1145/1409240.1409255.

Denso Wave. Qr code. Online. URL `http://www.denso-wave.com/qrcode/index-e.html`.

Alan Dix, Janet Finlay, Gregory D. Abowd, and Russell Beale. *Human-Computer Interaction*. Prentice-Hall Inc., 3 edition, 2004.

European Information Technology Observatory. More than four billion mobile phone users worldwide. Online, August

2009. URL `http://www.eito.com/pressinformation_20090807.htm`.

Beat Gfeller and Michael Rohs. Visual code recognition for camera-equipped mobile phones. Online, May 2006. URL `http://www.vs.inf.ethz.ch/res/proj/visualcodes/`.

Brent Hecht, Michael Rohs, Johannes Schöning, and Antonio Krüger. Wikeye - using magic lenses to explore georeferenced wikipedia content. In *Proceedings of the 3rd International Workshop on Pervasive Mobile Interaction Devices (PERMID*, 2007.

Niels Henze, Mingyu Lim, Andreas Lorenz, Michael Mueller, Xavier Righetti, Enrico Rukzio, Andreas Zimmermann, Nadia Magnenat-Thalmann, Susanne Boll, and Daniel Thalmann. Contextual bookmarks. *Mobile Interaction with The Real World*, 2007.

Niels Henze, René Reiners, Xavier Righetti, Enrico Rukzio, and Susanne Boll. Services surround you: Physical-virtual linkage with contextual bookmarks. *Vis. Comput.*, 24(7):847–855, 2008. ISSN 0178-2789. doi: http://dx.doi.org/10.1007/s00371-008-0266-4.

Iris Herbst, Anne-Kathrin Braun, Rod McCall, and Wolfgang Broll. Timewarp: interactive time travel with a mobile mixed reality game. In *MobileHCI '08: Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, pages 235–244, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-952-4. doi: http://doi.acm.org/10.1145/1409240.1409266.

Kaywa. Qr-code generator. Online. URL `http://qrcode.kaywa.com/`.

Kooaba AG. kooaba. Online, 2009. URL `http://www.kooaba.com/`.

Greg Kumparak. Google: Android now shipping on 60,000 handsets per day. Online, February 2010. URL `http://www.mobilecrunch.com/2010/02/16/google-now-shipping-60000-android-handsets-per-day/`.

makebarcode.com. Barcode comparison chart. Online, January 2010. URL `http://www.makebarcode.com/specs/barcodechart.html`.

Irene Mavrommati and Achilles Kameas. The evolution of objects into hyper-objects: will it be mostly harmless? *Personal Ubiquitous Comput.*, 7(3-4):176–181, 2003. ISSN 1617-4909. doi: http://dx.doi.org/10.1007/s00779-003-0223-1.

MobileInfo.com. Bluetooth frequently asked questions. Online, 2001. URL `http://www.mobileinfo.com/Bluetooth/FAQ.htm`.

Mobilizy. Wikitude world browser. Online, 2009a. URL `http://www.wikitude.org/`.

Mobilizy. Wikitude world browser. Online, May 2009b. URL `http://www.mobilizy.com/en/wikitude-ein-reisefuhrer`.

Mobilizy. Wikitude.me geo-tag the world. Online, 2009c. URL `http://www.wikitude.me/`.

Jakob Nielsen. Iterative user-interface design. *Computer*, 26 (11):32 – 41, 1993.

Donald A. Norman. *The Design of Everyday Things*. Basic Books, 2002.

Ohan Oda, Levi J. Lister, Sean White, and Steven Feiner. Developing an augmented reality racing game. In *INTETAIN '08: Proceedings of the 2nd international conference on INtelligent TEchnologies for interactive enterTAINment*, pages 1–8, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). ISBN 978-963-9799-13-4.

Celine Paloc, Eduardo Carrasco, Ivan Macia, Ruben Gomez, Inigo Barandiaran, Jose Manuel Jimenez, Oskar Rueda, Jorge Ortiz de Urbina, Andres Valdivieso, and Georgios Sakas. Computer-aided surgery based on autostereoscopic augmented reality. In *IV '04: Proceedings of the Information Visualisation, Eighth International Conference*, pages 189–193, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2177-0. doi: http://dx.doi.org/10.1109/IV.2004.41.

Ben Parr. Augmented wikipedia reality has arrived on the iphone. Online, October 2009. URL `http://mashable.com/2009/10/02/wikitude/`.

Perci. Perci (pervasive service interaction). Online. URL `http://www.hcilab.org/projects/perci/`.

René Reiners and Vina Wibowo. Prototyping an extended magic lens interface for discovering smart objects in a ubiquitous environment. In *IADIS AC (2)*, pages 313–316, 2009.

Bena Roberts. Mobile barcodes: Nfc vs qr? Online, January 2009. URL `http://www.gomonews.com/mobile-barcodes-nfc-vs-qr/`.

Suzanne Robertson and James Robertson. *Mastering the requirements process*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1999. ISBN 0-201-36046-2.

Michael Rohs and Beat Gfeller. Using camera-equipped mobile phones for interacting with real-world objects. In *Advances in Pervasive Computing*, pages 265–271, 2004.

Michael Rohs and Antti Oulasvirta. Target acquisition with camera phones when used as magic lenses. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1409–1418, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1. doi: http://doi.acm.org/10.1145/1357054.1357275.

Enrico Rukzio, Karin Leichtenstern, Vic Callaghan, Paul Holleis, Albrecht Schmidt, and Jeannette Chin. An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning. In *Lecture Notes in Computer Science - UbiComp 2006: Ubiquitous Computing*, volume 4206, pages 87–104. Springer Berlin, Heidelberg, September 2006. doi: 10.1007/11853565.

Enrico Rukzio, Gregor Broll, Karin Leichtenstern, and Albrecht Schmidt. Mobile interaction with the real world: An evaluation and comparison of physical mobile interaction techniques. In *Lecture Notes in Computer Science: Ambient Intelligence*, volume 4794/2007, pages 1–18. Springer Berlin, Heidelberg, November 2007. doi: 10.1007/978-3-540-76652-0_1.

Reinhard Sefelin, Manfred Tscheligi, and Verena Giller. Paper prototyping - what is it good for?: a comparison of paper- and computer-based low-fidelity prototyping. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 778–779, New York, NY, USA, 2003. ACM. ISBN 1-58113-637-4. doi: http://doi.acm.org/10.1145/765891.765986.

Semacode Corporation. Semacode. Online.

Alan D. Smith and Felix Offodile. Information management of automatic data capture: an overview of technical developments. *Information Management & Computer Security*, 10(3):109–118, 2002.

Carolyn Snyder. Paper prototyping, November 2001. URL `http://www.cim.mcgill.ca/~jer/courses/hci/ref/snyder.pdf`.

SPRX Mobile. Layar reality browser. Online, 2009a. URL `http://layar.com/`.

SPRX Mobile. Yes. we are going 3d! Online, September 2009b. URL `http://layar.com/3d/`.

Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. Comparative effectiveness of augmented reality in object assembly. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 73–80, New York, NY, USA, 2003. ACM. ISBN 1-58113-630-7. doi: http://doi.acm.org/10.1145/642611.642626.

United Nations Radio. More mobile phone users in 2010. Online, February 2010. URL `http://www.unmultimedia.org/radio/english/detail/90889.html`.

Daniel Wagner and Dieter Schmalstieg. History and future of tracking for mobile phone augmented reality. In *ISUVR '09: Proceedings of the 2009 International Symposium on Ubiquitous Virtual Reality*, pages 7–10, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-0-7695-3704-7. doi: http://dx.doi.org/10.1109/ISUVR.2009.11.

Mark Weiser. The computer for the 21st century. pages 933–940, 1995.

Peter C. Wright and Andrew F. Monk. The use of think-aloud evaluation methods in design. *SIGCHI Bull.*, 23(1): 55–57, 1991. ISSN 0736-6906. doi: http://doi.acm.org/10.1145/122672.122685.

Typeset 30 April 2010