Anàlisis de l'Interacció i Avaluació Heurística d'un Joc Multijugador per a Telèfons Mòbils

Departament d'Informàtica i d'Enginyeria Industrial

Universitat de Lleida – Lancaster University

Setembre del 2007

Memòria del Treball de Final de Carrera desenvolupat per Rubén Sarlé Laplana i dirigit pel Dr. Toni Granollers Saltiveri de l'Universitat de Lleida i el Dr. Alan Dix de l'Universitat de Lancaster.

Summary

This final year project has been carried out between the GRIHO Research Group of the Computing and Industrial Engineering Department of the *Universitat de Lleida* and the Interaction Group of the Computing Department of the Lancaster University.

This project is focused on an analysis and an heuristic evaluation of a multi-player game designed for mobile phones and based on an adaptation of Nielsen's and Molich's heuristics which was carried out by a group of researchers of the Lancaster University.

Rubén Sarlé Laplana
Lleida, Setembre 2007

Acknowledgements

I would like to thank all those people who have been helping me and who have given me the chance to study on this really interesting project.

First of all, I would like to thank the promoters of this project, Toni Granollers Saltiveri and Alan Dix for giving me the change to do my project at the Lancaster University and guiding me whenever needed. It has been an excellent opportunity and experience from which I have learnt many things. I would like to thank both Universities for giving me this great opportunity.

Secondly, I would like to thank my GRHIO class mates for their support. They have given me their friendship for both bad and good moments. I have learnt lots of things from you! Thanks a lot!

I also would like to give special thanks to Anna for being with me in this great experience, for helping and giving me all her support.

Finally, I would like to thank my family to whom this project is dedicated; my parents Jordi and Rosa and my sister Maria. Thanks for being the greatest support and for giving me the strength needed to achieve all my aims and dreams.

Index

1. INTRODUCTION ... 9

1.1. JUSTIFICATION AND WORK CONTEXT ... 9
1.2. INTRODUCTION ... 11
1.3. OBJECTIVES ... 12

2. HUMAN COMPUTER INTERACTION ... 13

2.1. DEFINITION ... 13
2.2. THE GOALS OF THE HCI ... 16
2.3. RELATIONSHIP OF HCI TO OTHER DISCIPLINES ... 16
2.4. SCENARIOS ... 17
2.4.1. Benefits ... 18
2.4.2. Method ... 18

3. USABILITY ENGINEERING ... 19

3.1. DEFINITION ... 19
3.2. THE USABILITY ENGINEERING LIFECYCLE ... 20
3.3. METHODS ... 20
3.4. CENTRED USER DESIGN ... 21
3.5. MOBILE DEVICES, APPLICATIONS AND THEIR CONTEXT ... 22
3.5.1. Limits posed by the device ... 22
3.5.2. Limits posed by context and interaction ... 23
3.6. MOBILE USABILITY ISSUES ... 24

4. HARDWARE AND SOFTWARE ... 26

4.1. HARDWARE ... 26
4.1.1. BRF 6300 ... 26
4.1.2. OMAP 3630 ... 27
4.1.3. WLAN ... 27
4.1.4. GSM/GPRS ... 28
4.2. SOFTWARE ... 28
4.2.1. Symbian ... 28
4.2.2. S60 ... 29
4.2.3. JAVA ... 29
4.3. BLUETOOTH OVERVIEW ... 30
4.3.1. Connection Setup ... 31
4.4. PICONET AND SCATTERNET ... 32
4.4.1. Creating a Connection ... 33
4.5. BLUETOOTH PROTOCOLS ... 36
4.5.1. L2CAP ... 36
4.5.2. RFCOMM ... 36

5. PLAY BY BLUETOOTH ... 37

5.1. GAMES ONCE STARTED ... 37
5.2. GAME DATA EXAMPLES ... 37
5.3.1. Description for two players ... 38
5.4. GAME SHUTDOWN ... 41
5.5. LOW-POWER MODE ... 41
5.6. DISCONNECTION AND LINK LOSS ... 41

6. SOFTWARE ANALYSIS ... 42

6.1. INTRODUCTION ... 42
6.2. HOST – CLIENT ROLES ... 42
6.3. HOST ROLE ... 43
6.4. CLIENT ROLE ... 43
6.5. CLIENT AND HOST ROLE ... 44
6.6. SCENARIOS ... 44

7. HEURISTIC EVALUATION 51

7.1. Introduction 51
7.2. Nielsen Heuristic Evaluation 52
7.2.1. Strengths of Heuristic Evaluation 53
7.2.2. Limitations of Heuristic Evaluation 53
7.3. Mobile Usability Heuristics 54

8. EVALUATION 56

8.1. Introduction 56
8.2. Punctuation Method 56
8.3. Evaluation 57
1. Visibility of system status and losability/findability of the mobile device 57
2. Match between system and the real world 58
3. Consistency and mapping 60
4. Good ergonomics and minimalist design 63
5. Ease of input, screen readability and glancability 64
6. Flexibility, efficiency of use and personalization 65
7. Aesthetic, privacy and social conventions 66
8. Realistic error management 66
8.4. Results 67

9. CONCLUSIONS 68

APPENDIX A 70

APPENDIX B 71

10. BIBLIOGRAPHY 72

Chapter 1: Introduction 72
Chapter 2: Human Computer Interaction 72
Chapter 3: Usability Engineering 73
Chapter 4: Hardware and Software 73
Chapter 5: Play by Bluetooth 74
Chapter 6: Software Analysis 74
Chapter 7: Heuristic Evaluation 74
Chapter 8: Evaluation 74
Appendix A 75
Appendix B 75

Figure Index

Figure 1: Sums up graphically all aspects related to the HCI 13
Figure 2: Lifecycle 20
Figure 3: Mobile Usability Issues 25
Figure 4: Smartphone application 26
Figure 5: BRF 6300 26
Figure 6: OMAP 3630 27
Figure 7: WLAN 28
Figure 8: (a) Point-to-Point (b) Point-to-Multipoint piconets 33
Figure 9: Scatternets 33
Figure 10: Point-to-Point Connection 34
Figure 11: Point-to-Multipoint Connection 34
Figure 12: Failed scatternet initiation to Host 35
Figure 13: Failed scatternet initiation to Client 35
Figure 14: Data exchange sequence chat 40
Figure 15: Actors 45
Figure 16: Interaction between Host and Client 47
Figure 17: Interaction between Player and Client 49
Figure 18: Process diagram of cards game 50
Figure 19: Severity Ranking Scale (SRS) 56
Figure 20: Main screen of the Operating System Symbian S60 57
Figure 21: Paired devices 58
Figure 22: Settings menu 58
Figure 23: Main menu, Connectivity menu, Bluetooth menu 59
Figure 24: Emulator of NOKIA with Symbian S60 OS 59
Figure 25: NOKIA 3250 70
Figure 26: Play Station Portable (Sony) 71

1. Introduction

1.1. Justification and Work Context

The idea of this research begun two years ago when I spoke with Toni Granollers Saltiveri, Professor in the Department of Computing at the University of Lleida. Toni recommended me to go to the Lancaster University because they are specialized in HCI[1] which is the subject I am going to work on. This University is giving me the chance to learn new fields and it is going to be a great honour to work with them.

First of all, I would like to introduce Alan Dix who is a Professor in the Department of Computing at Lancaster University. His interests include delays and temporal issues, interaction in mobile and ubiquitous systems, the use of formal methods in HCI, the design of cyberspace and the way information is transforming economics and society, bits of e-learning, etc.

Secondly, considering that this research is going to be written in Lancaster I would like to introduce you the city. It is a small city (population 45.952) situated in Lancashire County in the North-West of England, bounded to the west by the Irish Sea. It is a commercial, cultural and educational centre. Also it is vibrant student-friendly city. Just ten minutes by bus to the University it is filled with pubs, clubs, theatres, market sand shops.

Most of the city centre is pedestrianised and in addition to the usual high street retailers, there is a shopping arcade and a large indoor market which contains a fish market selling fresh local catches from Morecambe Bay.

Threading through the city is the canal with its waterside inns where you can watch the procession of boats glide by. There are punts for hire and a choice of craft offering cruises and restaurant facilities.

In the South, not far from the city centre, we find the prestigious University of Lancaster which won an international recognition for the quality of its teaching and research. The University provides an outstanding and welcoming students experience on a campus in a beautiful setting and in a safe and supportive environment.

The Computing Department at Lancaster University is one of the leading Computer Science Departments in the UK (and indeed Europe). All academic staff members at Lancaster are actively engaged in advanced research and they have a large and thriving community of research students from UK, from other European countries and overseas.

[1] Human Computer Interaction

In 2004, they have moved into InfoLab21, a brand new state-of-the-art research facility with the broader goal of creating a world-class research, development and business centre in Information and Communication Technologies. There are 250 researchers and it is one of the largest and most significant ICT[1] establishments worldwide.

Research at InfoLab21 is organised as a series of closely related and heavily over-lapping themes:

- Networks & distributed systems
- Mobile & ubiquitous computing
- Software systems engineering
- Cooperative and interactive system

Collectively, this provides comprehensive coverage of the major technologies and techniques underpinning contemporary computer systems from low level networking aspects to HCI issues.

HCI and interaction-related research at Lancaster covers a range of areas, some at the 'edge' of the discipline and some more traditional. They are a world leader in the technical and interaction issues of mobile and ubiquitous computing, and have long been the hub of work using ethnographic techniques in system design. Current research areas include wearable, CSCW[2], infrastructure for ubiquitous applications, situated displays, use of SMS[3] technology, applications in healthcare, location-aware devices and interfaces, multimedia, domestic environments, user experience, formal aspects of HCI.

To complement this, research staff also works on advanced software systems engineering tools and techniques to support the construction of complex systems with particular emphasis on understanding the social and human dimensions.

Collectively, this provides comprehensive coverage of the major technologies and techniques underpinning contemporary ICT[4] systems from lower level aspects such as wireless communication and digital signal processing through network protocols and middleware to emerging application domains and also consideration of human computer interaction.

Finally, I would like to mention the CASIDE[5] Project, created by the UK Government’s leading funding agency for research and training in engineering and the physical sciences. The main aim of this project is to understand the way in which the physical placement and design of

[1] Information and Communication Technologies
[2] Computer Supported Cooperative Work
[3] Short Message Service
[4] Information and Communication Technologies
[5] Investigating Cooperating Applications in Situated Display Environments

networked displays in semi-wild settings influences and facilitates coordination and community. This fundamental understanding will inform the development of suitable guidelines and methods for the design of situated displays both within and beyond the lifetime of the project.

1.2. Introduction

According to an article published the 7th of August of this year in *La Cadena SER*[1] more than 272,7 million of mobiles phones have been sold during the second quarter, almost 7% more than the previous quarter and 16,5% more than 2006; figure that shows both the evolution and the economic importance of this technology.

As users, we already know that mobile phones offer a wide range of services such as messaging, call services, MP3, video functions, digital video and camera as well as the execution of some games in Java format. But what is really interesting is the fact that Multinational companies specialized on this technology are introducing new services and functionalities to turn mobile phones into a leisure and entertainment tool.

As hardware and software technologies advance, new advanced functions are going to become available. Both software and hardware are closely tied to the computational development and they are adaptations of analysis procedures and programming languages used exclusively for the development of computer programmes.

The aim of this project is to study in depth some of the issues which surround the use of mobile phones as a part of group interactions with public displays, in particular analyzing a multi-player game, taking as an example a card game which explores issues of audience viewing on a large screen combined with personal per-player views on mobile phones.

Public displays are becoming ubiquitous ranging from Weiser's 'yard-scale' plasma screens in airports, hotels and public bars to tennis-court-scale urban screens and Lancaster is currently installing eCampus a campus-wide infrastructure of public displays. A key problem with these screens is that they are largely broadcast based, with little personal interaction.

The proposed work is part of a wider vision where we see phones and public displays of all sizes as part of a single interaction environment. Some of our other in-progress and planned work is focused on enhancing more informational aspects of displays including leaving messages in semi-public displays, influencing pre-vetted information for 'broadcast' style displays, and book marking and obtaining extra information about current topics on public displays.

[1] Spanish Radio (www.cadenaser.com)

This project is focused on the use of mobile phones as a display of games execution. Imagine yourself in a café or bar; the screen (used at other times for TV or newsfeed) switches to game mode and players connect using Bluetooth. The game is competitive perhaps star ship fighters, a car-racing or cards game. Players see a private heads-up view on their phone screen whilst the big screen shows a bird-eye overview of the whole game. In addition the big screen includes an area where a high fidelity '\movie' is shown of the game play, sometimes over the shoulder, sometimes mid-distance, where camera angles are chosen algorithmically based on 'hot spots' in the game (dogfight, overtaking on a corner). So, those who are not playing get a view that is not unlike a live sports event.

1.3. Objectives

The aim of this project is to advance in the integration and adaptation of computational procedures for mobile devices, analysing the software of a multi-player game and carrying out a Heuristic evaluation.

The main aspects of this project are the following ones:

- **Introduction of the HCI adapted to the mobile phone technology:** in chapter II the main aspects of the HCI from the mobile phone technology point of view are going to be analysed.
- **Usability analysis:** in chapter III an study of both the Usability Engineering and the Centred User Design is going to be carried out determining mobile technology limitations and mobile usability issues.
- **Hardware and software:** in chapter IV, an introduction of the hardware and software applied to the mobile technology and also the technical characteristics of Bluetooth connexions are going to be carried out.
- **Bluetooth analysis:** in chapter V the Bluetooth connection during a game execution is going to be carried out, analysing the scenario between two players.
- **Software analysis:** in chapter VI I am going to analyse on the one hand the host and client role and on the other hand the software, focusing on a multi-player card game. In order to see the relation between actors and scenarios.
- **Heuristic evaluation:** in chapters VII and VIII I am going to introduce the Heuristic Evaluation (Nielsen and Molich), also heuristics for mobile devices and finally, an example of a heuristic evaluation based on the mobile technology.

2. Human Computer Interaction

2.1. Definition

Human Computer Interaction, or HCI, is the study, planning, and design of what happens when you and a computer work together. As its name implies, HCI consists of three parts: the user, the computer itself, and the ways they work together.

According to the Special Interest Group in Computer Human Interaction (SIGCHI) of the Association for Computer Machinery (ACM), the Human Computer Interaction is defined as follows:

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.

The definition of Human Computer Interaction is understood as an interaction between humans and computer systems. The illustration shown below it is an adaptation of the general interaction rules to the interaction between humans and mobile phones.

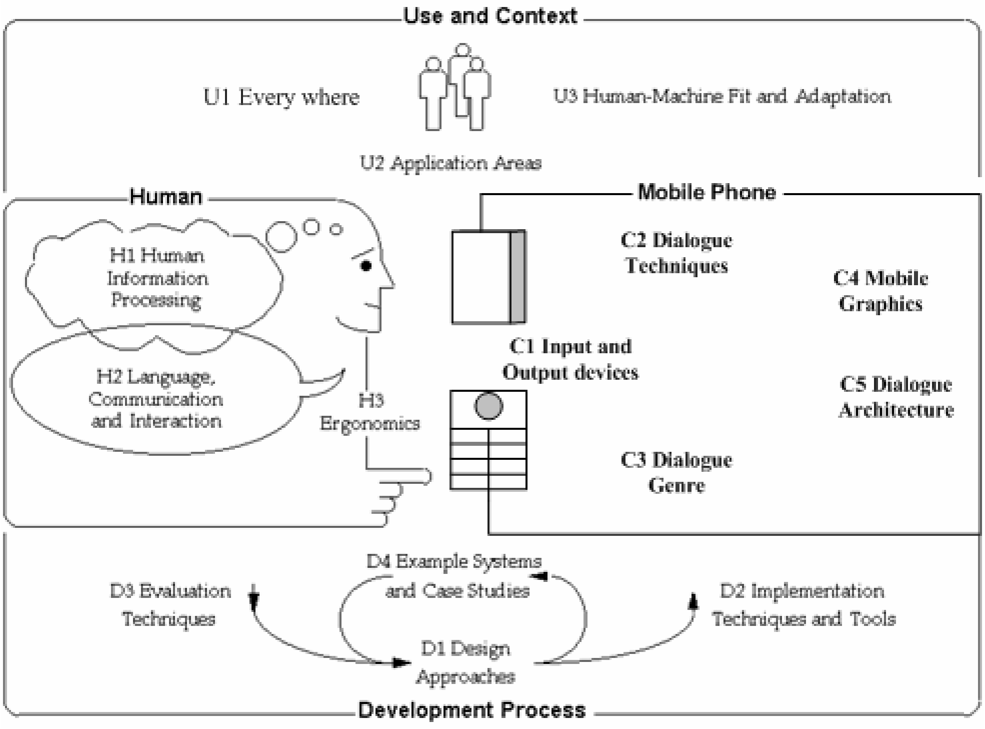


Figure 1: Sums up graphically all aspects related to the HCI

The User

When we talk about HCI, we don't necessarily imagine a single user with a mobile phone. By "user", we may mean an individual user, a group of users working together, or maybe even a series of users in an organisation, each involved with some part of the job, development or a game. The user is whoever is trying to get the job done using the technology. An appreciation of the way people's sensory systems (sight, hearing, touch) relay information is vital to designing a first-class product. For example, display layouts should accommodate the fact that people can be sidetracked by the smallest movement in the outer (peripheral) part of their visual fields, so only important areas should be specified by moving or blinking visuals. And of course, people like designs that grab their attention. Designers must decide how to make products attractive without distracting users from their tasks.

The Mobile Phone

When we talk about mobile phones, we are referring to any technology ranging from PDAs to a large scale of mobile technology systems, even a process control system or an embedded system could be classed as mobile technology.

Interacting with the physical world via a mobile handheld device is a relatively new paradigm, which has quickly emerged during recent years. Several factors have contributed to this development trend. Technological advances in mobile devices such as component miniaturization and diminished energy consumption have enabled development of gadgets that have more features and computing power. Integrating cameras, motion sensors, and radio frequency identification (RFID) or barcode readers have made new interaction concepts possible. Mobile devices have become extremely common, as mobile phone adoption rate approaches 100% of the population in numerous countries. Moreover, a mobile phone is typically carried with the owner, and thus offers a personal computing platform that is practically constantly accessible by the individual user. Due to these factors, mobile phones hold much potential as a platform for accessing ubiquitous computing services and interacting with smart environments.

The Interaction

There are obvious differences between humans and machines. In spite of these, HCI attempts to ensure that they both get on with each other and interact successfully. In order to achieve a usable Website, you need to apply what you know about humans and computers, and consult with likely users throughout the design process. You need to find a reasonable balance between what can be done within the schedule and budget, and what would be ideal for your users.

The context

The word Context in general use has a multitude of meanings. Even within the field of computer science different disciplines, such as artificial intelligence, natural language processing, image recognition, and more recently mobile computing, have their very own understanding of what context is. In our work we found that very general descriptions of context as given by a dictionary and also synonyms found in a thesaurus come very close to our understanding. To illustrate this we like to provide the following definitions:

Context n 1: *discourse that surrounds a language unit and helps to determine its interpretation [syn: linguistic context, context of use]* ***2:*** *the set of facts or circumstances that surround a situation or event; "the historical context" (Source: WorldNet ® 1.6)*

Context: *That which surrounds, and gives meaning to, something else. (Source: The Free On-line Dictionary of Computing)*

Synonyms Context: *Circumstance, situation, phase, position, posture, attitude, place, point; terms; regime; footing, standing, status, occasion, surroundings, environment, location, dependence.*

In our work we propose to regard situational context, such as location, surrounding environment or state of the device, as implicit input to the system. We use the term situational context to describe implicit interaction fragments. This extends the concept of context beyond the informational context into real world environments.

The New Paradigm: Ubiquitous Computing

Ubiquitous computing, or calm technology, is a paradigm shift where technology becomes virtually invisible in our lives. Instead of having a desk-top or lap-top machine, the technology we use will be embedded in our environment. From the ubiquitous computing page at Xerox PARC [UBPARC] we have the following description: imagine a world with hundreds of wireless computing devices of different sizes in the same room. In order to bring this type of computing out into the environment, among the things we need to rethink are user interfaces, displays, operating systems, networks, and wireless communications. Contemporary devices that lend some support to this latter idea include mobile phones, digital audio players, radio-frequency identification tags and interactive whiteboards.

Ubiquitous computing is changing our daily activities in a variety of ways. When it comes to using today's digital tools users tend to:

- communicate in different ways
- be more active
- conceive and use geographical and temporal spaces differently
- have more control

In addition, ubiquitous computing is:

- global and local
- social and personal
- public and private
- invisible and visible
- an aspect of both knowledge creation and information dissemination

2.2. *The goals of the HCI*

The goals of HCI are to produce usable and safe systems, as well as functional systems. In order to produce mobile phone systems with good usability, developers must attempt to:

- **understand** the factors that determine how people use technology
- **develop** tools and techniques to enable building suitable systems
- **achieve efficient**, effective, and safe interaction

Underlying the whole theme of HCI is the belief that people using a mobile phone system should come first. Their needs, capabilities and preferences for conducting various tasks should direct developers in the way that they design systems. People should not have to change the way that they use a system in order to fit in with it. Instead, the system should be designed to match their requirements.

2.3. *Relationship of HCI to other disciplines*

HCI is known as a multi-disciplinary field – what this means is that HCI draws expertise from a number of different areas of study. It needs to do this because HCI designers are required to understand human beings – their behaviour and mental capacities as well as their needs in organisational settings. Designers also need to know about computer systems and software.

Some of the disciplines that contribute to HCI are:

- ergonomics - the study of the design of work
- computer science
- artificial intelligence
- linguistics
- psychology
- sociology
- software engineering
- design

A student of HCI will not need to know all about these other subjects in depth, of course. However, it is important to be aware that in HCI, we may have to use the knowledge from some of these other disciplines to solve a problem in a certain situation.

2.4. Scenarios

A scenario is a specific example of an individual user interacting with an information system. It is used within a design study to structure and communicate information about how a design might be used in a real world situation. The scenario describes what the user is trying to do and the method by which he or she would go about completing it.

When beginning an interface design and trying to imagine how the software will work or look is often the most difficult step in the whole design process. Many designers have difficulty in knowing where to start. One of the best ways is to create a description or scenario of the way the software or user interface will work.

A scenario can be a ‘snapshot’ of the system in use, or a more general description of the users and their goals, or the basic functions of the system. You must decide how specific to be depending on the needs of your project. A scenario should consider the general user interface, including the usefulness of the system, the environment the system will be in, and the tools for interaction (such as a mouse-driven personal computer, a touch-screen wall-mounted information system or a wall accessed audio system). Writing a scenario helps you think about all of these aspects of your user interface, and helps you answer some questions about your design. In many ways, writing a scenario is like writing a story about your users to help you identify many of their characteristics.

2.4.1. Benefits

- It encourages designers to consider the characteristics of the intended users, their tasks and their environment.
- Usability issues can be explored at a very early stage in the design process (before a commitment to code has been made).
- Scenarios can help identify usability targets and likely task completion times.
- The method promotes developer buy-in and encourages a user-centred design approach.
- Scenarios can also be used to generate contexts for evaluation studies.
- Only minimal resources are required to generate scenarios.
- The technique can be used by developers with little or no human factors expertise.

2.4.2. Method

An experienced moderator is recommended for the sessions in which the scenario is explored.

- Gather together the development team and other relevant stakeholders under the direction of an experienced facilitator.
- Identify intended users, their tasks and the general context. This information will provide the basis for the scenarios to be created by the development team.
- Functionally decompose user goals into the operations needed to achieve them.
- Consider which activities should be performed by the user and which by the computer.
- Create an outline of the users' activities, goals and motivations for using the system being designed, and the tasks they will perform.
- To maintain design flexibility, scenarios should not specify what product features are used.
- Assign task time estimates and completion criteria as usability targets.
- The session can be videotaped for later review or transcribed for wider distribution.
- The results from scenario building sessions can be used to plan user-based evaluations.

3. Usability Engineering

3.1. Definition

It is important to realise that usability is not a single, one-dimensional properly of a user interface. Usability has multiply components and is traditionally associated with these five usability attributes:

- **Learnability:** The system should be easy to learn so that the user can rapidly start getting some work done with the system.

- **Efficiency:** The system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible.

- **Memorability:** The system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again.

- **Errors:** The system should have a low error rat, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them. Further, catastrophic errors must not occur.

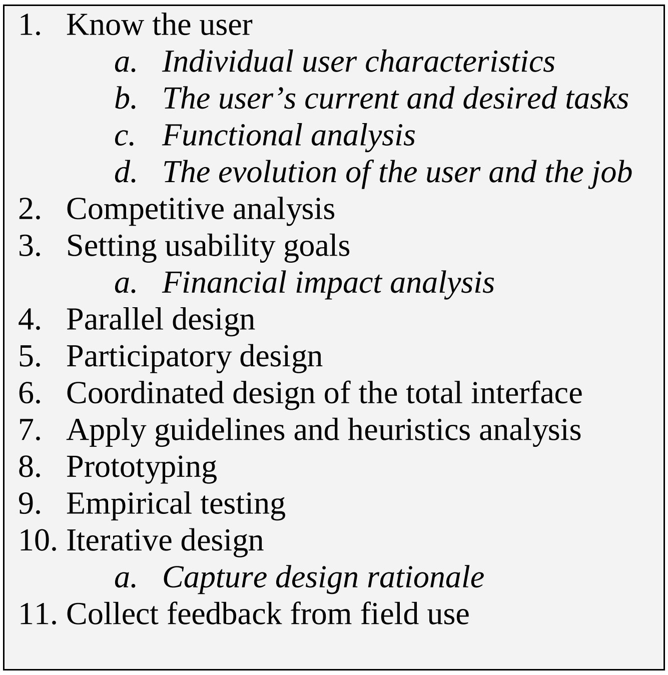
- **Satisfaction:** The system should be pleasant to use, so that users are subjectively satisfied when using it; they like it.

Usability is typically measured by having a number of test users (selected to be as representative as possible of the intended users) use the system to perform a prespecified set of tasks, though it can also measure by having real users in the field perform whatever tasks they are doing anyway. In either case, an important point is that usability is measured relative to certain users and certain tasks. It could well be the case that the same system would be measured as having different usability characteristics if used by different users for different tasks.

To determine a system's overall usability on the basis of a set of usability measures, one normally takes the mean value of each of the attributes that have been measured and checks whether these means are better than some previously specified minimum. Since the users are known to be very different, it is probably better to consider the entire distribution of usability measures and not just the mean value.

3.2. The Usability Engineering Lifecycle

Usability engineering is not a one-short affair where the user interface is fixed up before the release of a product. Rather, usability engineering is a set of activities that ideally takes place throughout the lifecycle of the product, with significant activities happening at the early stages before the user interface has ever been designed. The need to have multiple usability engineering stages supplement each other was recognized early in the field, though not always followed on development projects.



1. Know the user
    a. *Individual user characteristics*
    b. *The user's current and desired tasks*
    c. *Functional analysis*
    d. *The evolution of the user and the job*
2. Competitive analysis
3. Setting usability goals
    a. *Financial impact analysis*
4. Parallel design
5. Participatory design
6. Coordinated design of the total interface
7. Apply guidelines and heuristics analysis
8. Prototyping
9. Empirical testing
10. Iterative design
    a. *Capture design rationale*
11. Collect feedback from field use

Figure 2: Lifecycle

3.3. Methods

There are a variety of approaches to usability evaluation that you may choose to take. The methodologies can be divided into two broad categories: those that gather data from actual users and those that can be applied without actual users present.

Your choice of method depends on:

- Cost of evaluation
- Appropriateness to project
- Time constraints
- Cost of implementation
- Cost of training new users

Usability evaluations can be conducted at many stages during and after the design and development process. In choosing a method, it is important to calculate the cost not only in terms of time and materials involved, but also in terms of the impact on the end-users, especially considering the cost of losing return visitors to your website due to unusable design.

- **Cognitive Walkthrough** is an approach to evaluating an interface based on breaking down and analyzing actions that a user must perform in order to use the system or perform a task.
- **Focus Groups** gather groups of users to get their feedback, initial reactions to a design, and discuss their preferences. Focus groups can be useful for raising issues that may not come out during interviews.
- **GOMS** is a family of techniques for modelling and describing human task performance. GOMS is an acronym that stands for Goals, Operators, Methods, and Selection Rules.
- **Prototyping** involves developing representations of a system for testing purposes and can range from simple sketches to almost fully functional systems.
- **Task Analysis** evaluates how the end-user actually uses software or websites. An analyst determines the user goals and tasks, then makes recommendations aimed at increasing efficiency and user-friendliness.
- **Usability Inspection** reviews a system based on a set of usability guidelines. Experts familiar with issues of usability in design perform the usability inspection.
- **User Testing** observes actual users interacting with software or websites. Users are asked to perform tasks while usability experts observe and take note of their actions.

3.4. Centred User Design

ISO 13407 is a description of best practice in user centred design. It provides guidance on design activities that take place throughout the life cycle of interactive systems. It describes an iterative development cycle where product requirements specifications correctly account for user and organisational requirements as well as specifying the context in which the product is to be used.

Design solutions are then produced which can be evaluated by representative users, against these requirements. The goal of the standard is to ensure that the development and use of interactive systems take account of the needs of the user as well as the needs of the developer and owner... to name but a few stakeholders.

The standard applies to software products, hardware/software systems, websites, and services.

The Standard describes:

Four Principles of Human-Centred Design:

- active involvement of users
- appropriate allocation of function to system and to user
- iteration of design solutions
- multi-disciplinary design

Four Human-Centred Design Activities:

- understand and specify the context of use
- specify user and organisational requirements
- produce more than one candidate design solution
- evaluate designs against requirements

Human-centred processes take account of context of use, the complete environment in which the interactive system will be used. The cooperative nature of so much computer related activity, whether in the office or the home, must inform the design of the systems. Whereas the scope of usability style guides for example, might be restricted in the main to screen design, human centred processes deal with the total system within which software and hardware are components. A human-centred approach to process modelling and re-engineering, addresses cultural issues and staff acceptance and thus encourages buy-in to new processes, a prerequisite for institutionalising process improvement in any organisation.

3.5. Mobile devices, applications and their context

In order to better understand how usability in mobile computing can be evaluated and improved, it is useful to outline specific limitations inherent in mobile devices, applications and their context. These falls into two broad categories: limitations due to the nature of the devices themselves, and limitations due to context of use and style of interaction.

3.5.1. Limits posed by the device

- ***Small-screen*****:** In order to be portable, mobiles must necessarily be small, and tend to have small screens, therefore problems due to the screen real estate are intrinsic and can be addressed only by figuring out new techniques to organize information visually.

- ***Limited input*****:** Because of device format, input mechanisms are inherently limited. Currently the most common means of interaction are: numeric keypads, which are used in almost all cell phones; and styluses, which are the primary input means in PDAs and smart phones.

- ***Limited bandwidth and cost*****:** Mobile Internet connections are still slow. This is in fact still one of the main factors limiting mobile Internet access. To this we must also add the problem of the cost model. Most companies offer their Internet access in a pay per KByte policy that obviously limits the size of pages and the number of requests.

- ***Limited connectivity*****:** Perhaps more than bandwidth, the latency of the connection affects its usability. The limited coverage of different networks and the consequent intermittent connection makes the latency extremely variable, as well as giving rise to problems of how to portray these hidden network properties to the user. There is also the problem of seamlessly switching between different types of network, e.g. WiFi to GPRS.

- ***Limited computational resources*****:** This means the capabilities of applications are limited. However, this should be overcome in the near future as new processors and memories specifically designed for mobile devices increase their quality and speed.

- ***Limited power*****:** This is often an underestimated issue, but the batteries are still a big problem for every kind of mobile system (laptops included). This has a big impact on end users: limited autonomy means limited availability, which in turn means limited reliability.

- ***Wide heterogeneity*****:** Users of mobile systems must always adapt to new forms of interaction as they switch to different mobiles. Changing the physical device and operating system usually translates into the need to re-learn functions, operations, messages, etc., with an enormous waste of resources.

3.5.2. Limits posed by context and interaction

- ***Variable context*****:** Since mobile devices, by definition, are mobile, the context in which they are used is continually changing. This poses challenging new issues because, though context has always been considered a fundamental aspect to analyse in usability studies, only now must we address such frequent and complex context variations within the same device, application, or single user.

- ***Kind of interaction*****:** The nature of interaction also changes in mobile settings. In general, users tend to interact in small and focused chunks of activities, more so than in fixed settings. A high proportion of tasks in a mobile environment consist of a few fast steps that the user should be able to execute without cognitive effort. In addition, mobile tasks may happen in conditions where users' attention is necessarily reduced, or may be part of more complex activities with which they should not interfere.

- ***Interruptions*:** Mobile devices/applications are always "with us". If this, on one hand, means that computation and data are always available, it is also true that notifications and requests for attention can happen at inappropriate moments and that some tasks may be interrupted. This raises two kinds of problem: appropriateness of notifications and recovery from interruptions.

- ***Privacy and security*:** Privacy issues become more prominent. While staying mobile, users find themselves in a variety of spaces (private and public), in a variety of situations (formal and informal), and in a variety of infrastructures (wireless and cable connection). Moving through these settings means having different needs for privacy and security.

- ***Intimacy and availability*:** Because mobile devices are mobile, they are personally available in a way that fixed devices are not. Moreover, they seem to engender a sense of being "personal" in a deeper sense than desktop PCs.

3.6. Mobile Usability Issues

The mobile usability issues are defined and classified in the following points:

- ***Interaction with device and infrastructure*:** many of the problems we have found in our research have a strong connection with the limits of the device and/or the infrastructure it is connected to.

- ***Interaction with application*:** this collects classes of problems connected to traditional screen design and information architecture.

- ***Cognitive issues*:** here we characterize usability problems stemming from an overload of cognitive resources or a mismatch between a cognitive model and reality. While this aspect has always been taken into account in traditional studies, in mobile settings it becomes more evident and presents new challenges.

- ***Personalization*:** standard heuristics tend to overlook problems connected to personalization or adaptation. While in standard settings this issue can be considered minor with respect to others, with mobile devices this aspect can really be critical.

- ***Social issues*:** mobile devices and applications are used in a wide spectrum of environments and social conditions: private or public, alone or in groups, etc. This means that the social impact of adopted design solutions cannot be underestimated. Issues like privacy, security, user image, and social conventions thus become of great importance.

- ***Context*:** similarly to social issues, it is necessary to take into account how the environment can affect interaction. Not only do social conventions and relationships with people matter, but also how potential physical environment features affect the design of an interface.

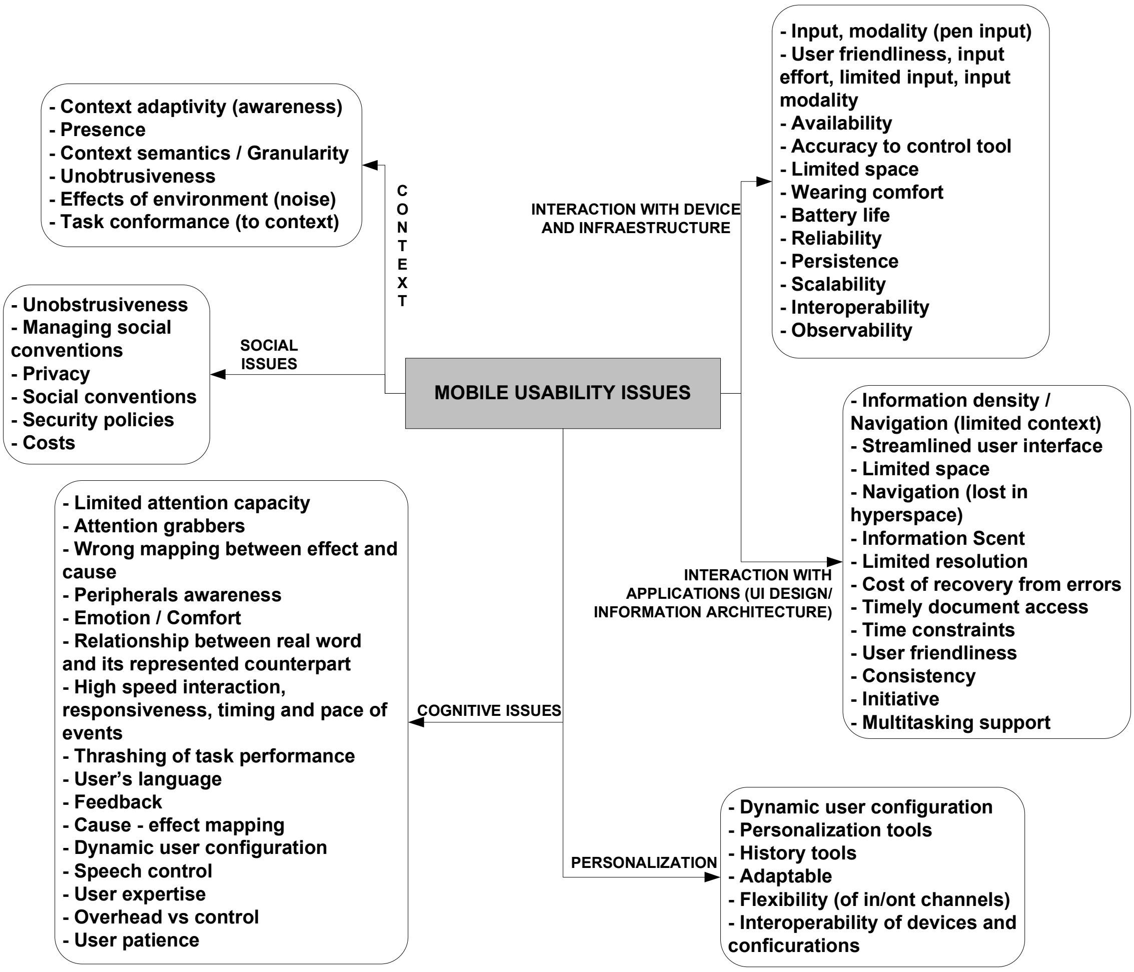

MOBILE USABILITY ISSUES
CONTEXT
- Context adaptivity (awareness)
- Presence
- Context semantics / Granularity
- Unobtrusiveness
- Effects of environment (noise)
- Task conformance (to context)
SOCIAL ISSUES
- Unobstrusiveness
- Managing social conventions
- Privacy
- Social conventions
- Security policies
- Costs
COGNITIVE ISSUES
- Limited attention capacity
- Attention grabbers
- Wrong mapping between effect and cause
- Peripherals awareness
- Emotion / Comfort
- Relationship between real word and its represented counterpart
- High speed interaction, responsiveness, timing and pace of events
- Thrashing of task performance
- User's language
- Feedback
- Cause - effect mapping
- Dynamic user configuration
- Speech control
- User expertise
- Overhead vs control
- User patience
INTERACTION WITH DEVICE AND INFRAESTRUCTURE
- Input, modality (pen input)
- User friendliness, input effort, limited input, input modality
- Availability
- Accuracy to control tool
- Limited space
- Wearing comfort
- Battery life
- Reliability
- Persistence
- Scalability
- Interoperability
- Observability
INTERACTION WITH APPLICATIONS (UI DESIGN/ INFORMATION ARCHITECTURE)
- Information density / Navigation (limited context)
- Streamlined user interface
- Limited space
- Navigation (lost in hyperspace)
- Information Scent
- Limited resolution
- Cost of recovery from errors
- Timely document access
- Time constraints
- User friendliness
- Consistency
- Initiative
- Multitasking support
PERSONALIZATION
- Dynamic user configuration
- Personalization tools
- History tools
- Adaptable
- Flexibility (of in/ont channels)
- Interoperability of devices and conficurations

Figure 3: Mobile Usability Issues

4. Hardware and Software

4.1. Hardware

Figure 1 shows a Smartphone application with its components.

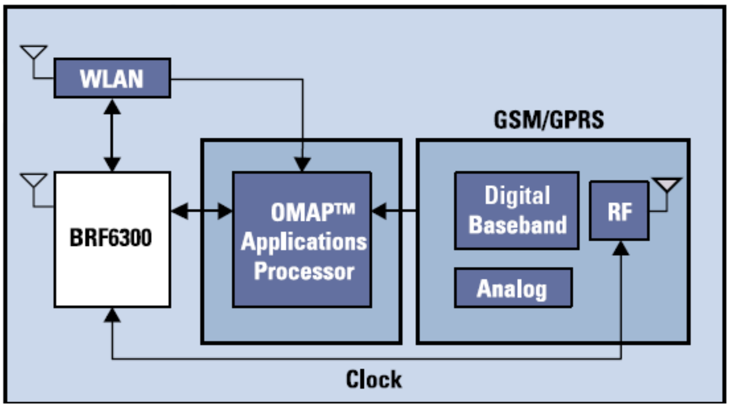


Figure 4: Smartphone application

4.1.1. BRF 6300

The BRF6300 from Texas Instruments (TI) is a highly integrated, digital CMOS, single-chip solution supporting Bluetooth® Specification v2.0. The new chip is based on TI's prior generation BRF6150, leveraging and exceeding its capabilities to provide maximum Enhanced Data Rate (EDR) support and lowest power consumption in most Bluetooth scenarios. The BRF6300 requires a small number of external components, offers enhanced WLAN co-existence algorithms and is software upgradeable to support future Bluetooth technology enhancements.

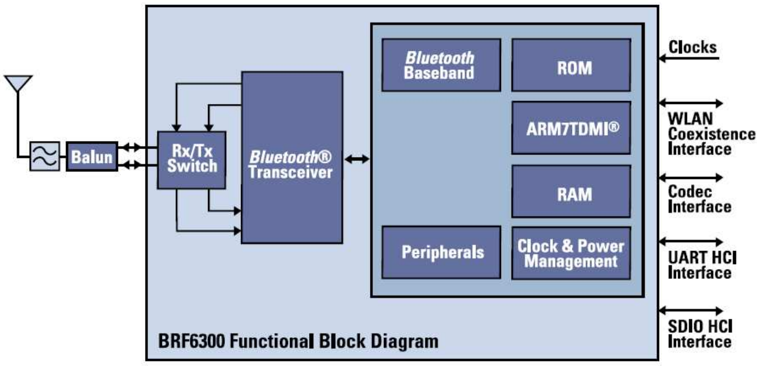


Figure 5: BRF 6300

4.1.2. OMAP 3630

The new OMAP3430 multimedia applications processor from Texas Instruments (TI) introduces a new level of performance that enables laptop-like productivity and advanced entertainment in 3G handsets. As the first product in TI's OMAP™ 3 family of applications processors with the ARM Cortex-A8 superscalar microprocessor core, the OMAP3430 delivers up to 3X gain in performance over ARM11-based processors. The new processor leverages industry-leading technologies to provide mobile phone battery life together with the performance needed for laptop-comparable productivity software and an audio-video experience equivalent to that of consumer electronics devices.

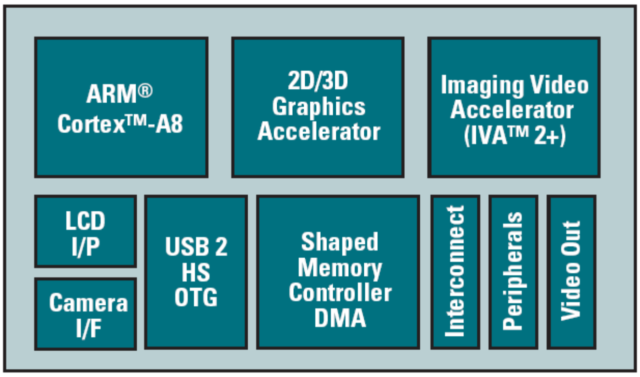

ARM®
Cortex™-A8
2D/3D Graphics Accelerator
Imaging Video Accelerator (IVA™ 2+)
LCD I/P
Camera I/F
USB 2 HS OTG
Shaped Memory Controller DMA
Interconnect
Peripherals
Video Out

Figure 6: OMAP 3630

4.1.3. WLAN

TI's WiLink™ 6.0 mobile platform is a complete hardware and software offering comprising proven, carrier-quality mobile WLAN, Bluetooth® and FM cores integrated onto a single chip.

There are two solutions in the WiLink 6.0 product offering. The WL1271 supports 802.11b/g, while the WL1273 supports 802.11a/b/g/n. Both single-chip solutions support Bluetooth Specification 2.1 + EDR and FM transmit and receive.

The WiLink 6.0 single-chip solutions are manufactured in 65-nm CMOS process and use TI's DRP™ technology to deliver low power, a small form factor and low cost requirements of handset manufacturers worldwide.

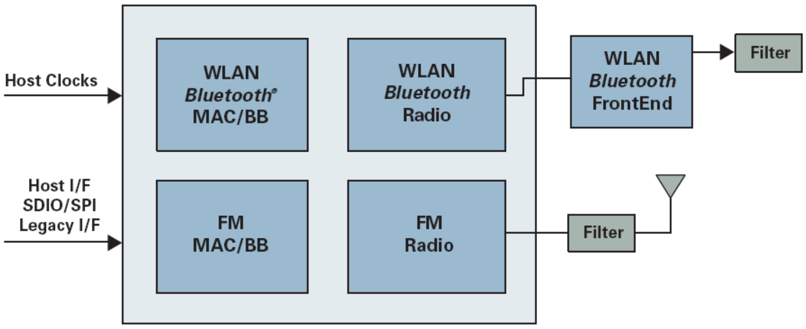


Figure 7: WLAN

4.1.4. GSM/GPRS

GPRS (General Packet Radio Service) is the world's most ubiquitous wireless data service, available now with almost every GSM network (Global System Mobile Communication, standard). GPRS is a connectivity solution based on Internet Protocols that supports a wide range of enterprise and consumer applications. With throughput rates of up to **40 kbit/s,** users have a similar access speed to a dial-up modem, but with the convenience of being able to connect from anywhere. GPRS customers enjoy advanced, feature-rich data services such as colour Internet browsing, e-mail on the move, powerful visual communications such as video streaming, multimedia messages and location based services.

For operators, the adoption of GPRS is a fast and cost-effective strategy that not only supports the real first wave of mobile Internet services, but also represents a big step towards 3GSM (or wideband-CDMA) networks and services.

4.2. Software

4.2.1. Symbian

Symbian OS is the advanced, open operating system licensed by the world’s leading mobile phone manufacturers. It is designed for the specific requirements of advanced 2.5G and 3G mobile phones. Symbian OS combines the power of an integrated applications environment with mobile telephony, bringing advanced data services to the mass market.

Symbian OS supports a wide range of device categories with several user interfaces, including Nokia S60, UIQ and the NTT DoCoMo common software platform for 3G FOMATM handsets. The commonality of Symbian OS APIs enables development that targets all of these phone platforms and categories.

4.2.2. S60

A major platform for Smartphone. Based on the Symbian operating system, S60 provides a standardized interface and a standardized platform for developers to create advanced applications for phones. S60 was originally developed as "Series 60" by Nokia. Nokia uses S0 for its own Smartphone, and also licenses the platform to other phone manufacturers, such as Siemens and Samsung. As with other major Smartphone platforms, S60 requires a large colour display, a powerful processor chip, a large amount of memory for storing applications and content, (usually expandable with memory cards,) and a standard set of keys for input and control. S60 phones can run applications and games written for the platform, as well as Java applications and games.

Unlike PDA-based platforms and certain other Symbian-based platforms, S60 does not require a touch screen or pen input, although touch-screen support is being worked on for future versions of S60.

4.2.3. JAVA

Java refers to a number of computer software products and specifications from Sun Microsystems (the Java™ technology) that together provide a system for developing and deploying cross-platform applications. Java is used in a wide variety of computing platforms spanning from embedded devices and cell phones on the low end to enterprise servers and super computers on the high end. Java is fairly ubiquitous in cell phones, Web servers and enterprise applications, and somewhat less common in desktop applications, though users may have come across Java applets when browsing the Web.

The Java platform is the name for a bundle of related programs, or platform, from Sun Microsystems which allow for developing and running programs written in the Java programming language. The platform is not specific to any one processor or operating system, but rather, an execution engine (called a virtual machine) and a compiler with a set of standard libraries are implemented for various hardware and operating systems so that Java programs can run identically on all of them.

Different editions of the platform are available, including:

- J2SE: Java 2 Standard Edition
- J2EE: Java 2 Enterprise Edition
- **J2ME: Java 2 Micro Edition (**simplify version of J2SE**)**

Java™ Platform, Micro Edition (Java ME) is the most ubiquitous application platform for mobile devices across the globe. It provides a robust, flexible environment for applications running on a broad range of other embedded devices, such as mobile phones, PDAs, TV set-top boxes, and printers. The Java ME platform includes flexible user interfaces, a robust security model, a broad range of built-in network protocols, and extensive support for networked and offline applications that can be downloaded dynamically. Applications based on Java ME software are portable across a wide range of devices, yet leveraging each device's native capabilities.

Simple JAVA application: Connected Limited Device Configuration or CLDC. Establishes the minimum hardware required for J2ME:

- 160kb of available for Java
- Processor de 16-bit
- Low energise
- Network connection (96000 bps or less)
- clock speed 16 Mhz or higher

The memory requirements are:

- 128kb available no volatile memory for MIDP (Mobile Information Device Profile) API bookshop.
- 32kb available volatile memory for Java runtime.
- 8kb available no volatile memory for dates of persistent applications.

KVM is the Virtual Machine smaller that Sun has to develop. For a standard configuration, require about 50 y 80 Kbytes of memory, the dynamic memory requirement is not more 180 Kbytes. Work with a processor of 16 o 32 bits.

4.3. Bluetooth Overview

Bluetooth wireless technology is a short-range (10 meters) communications technology intended to replace the cables connecting portable and/or fixed devices while maintaining high levels of security. The key features of Bluetooth technology are robustness, low power, and low cost. The Bluetooth specification defines a uniform structure for a wide range of devices to connect and communicate with each other.

Bluetooth technology has achieved global acceptance such that any Bluetooth enabled device, almost everywhere in the world, can connect to other Bluetooth enabled devices in proximity. Bluetooth enabled electronic devices connect and communicate wirelessly through short-range, ad hoc networks known as piconets. Each device can simultaneously communicate with up to seven other devices within a single piconet.

The Bluetooth core system consists of an RF[1] transceiver, baseband, and protocol stack. The system offers services that enable the connection of devices and the exchange of a variety of data classes between these devices. Some of the most important characteristics are:

- Power: The most commonly used 2.5 mW of power. Bluetooth technology is designed to have very low power consumption. This is reinforced in the specification by allowing radios to be powered down when inactive.

- DataRange: 1 Mbps for Version 1.2; Up to 3 Mbps supported for Version 2.0 + EDR[2]

- Secure Connections: From the start, Bluetooth technology was designed with security needs in mind. Since it is globally available in the open 2.4 GHz ISM[3] band, robustness was built in from the beginning. With adaptive frequency hopping (AFH), the signal “hops” and thus limits interference from other signals. Further, Bluetooth technology has built-in security such as 128bit encryption and PIN code authentication. When Bluetooth products identify themselves, they use the PIN code the first time they connect. Once connected, always securely connected.

4.3.1. Connection Setup

Bluetooth devices need to exchange information before actual data transmission. This information exchange involves inquiry, service discovery, and remote name request.

Inquiry

Since Bluetooth devices are mobile and form networks dynamically, they need a way to find other nearby devices to connect to. This process is called the inquiry procedure in the Bluetooth Baseband Specification. From the point of view of an application, inquiry means collecting Bluetooth addresses and determining the class of device (CoD) in the vicinity.

The CoD value indicates the capability of the device. The CoD field shows if a device has rendering capabilities, like a printer, or is an audio device, like a headset. Some numerical values are reserved by the Bluetooth specification.

For inquiry, an access code is used. There is one General Inquiry Access Code (GIAC) to inquire for any nearby Bluetooth devices, and a number of Dedicated Inquiry Access Codes (DIAC) that inquire only for a specific type of device. The inquiry access codes are derived from

[1] Radio Frequency
[2] Enhanced Data Rate
[3] Industrial, scientific and medical

reserved Bluetooth device addresses and are further described in the Bluetooth Baseband Specification.

Normally, inquiry is done with the GIAC. But the Java and Symbian Bluetooth APIs also offer the possibility of using a DIAC. In some papers, DIAC is also called Limited Inquiry Access Code (LIAC).

Service discovery

Inquiry is the first step; this is followed by service discovery, which is based on the Service Discovery Protocol (SDP).

In wired networks, there is normally a central infrastructure for managing connections; Bluetooth doesn’t provide this. Instead, connections are made dynamically, and devices must determine what services are provided on discovered devices. This is called service discovery, which is the process by which applications locate and gather information about other services in the vicinity.

Remote name request

Inquiry just provides the user with bulky BD addresses. To obtain user-friendly names for remote devices, the remote name request procedure is used.

4.4. Piconet and Scatternet

A basic Bluetooth network is a piconet. The device that invites other devices into a piconet becomes a Host, and a device that accepts such an invitation becomes a Client, although the roles can be switched later in some Bluetooth stacks. The Host role does not imply any privileges, but means that the Host device governs the baseband synchronization between devices. Figure 8 illustrates two kinds of piconet configurations, the first with a point-to-point connection (suitable for head-to-head games) and the second with point-to-multipoint connection (suitable for games with three to seven players). The Clients in the piconet only link to the Host; there are no direct links between the Clients.

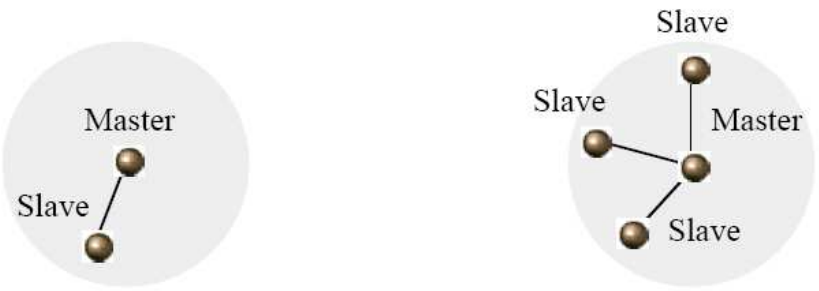


Figure 8: (a) Point-to-Point (b) Point-to-Multipoint piconets

The Bluetooth Baseband Specification limits the number of active Clients to seven in each piconet. It is also possible to establish a scatternet if the hardware and software stacks support it. Figure 9 illustrates two scatternet configurations.

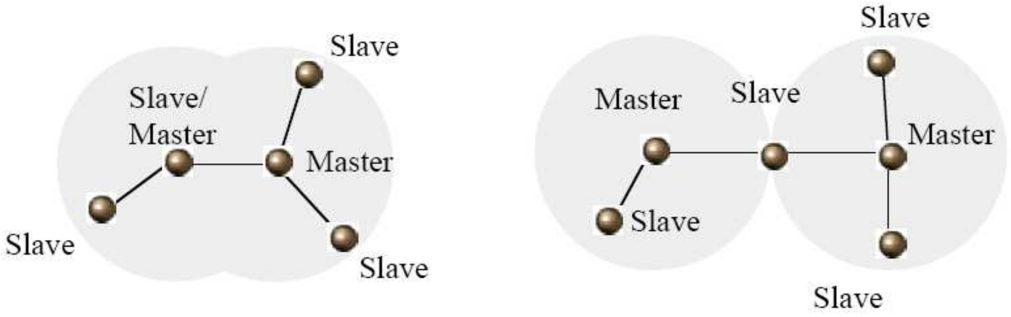


Figure 9: Scatternets

In a scatternet, the Host of one piconet can be the Client of another piconet. Or a Client can be a Client in two piconets. The disadvantage of scatternets is that the piconets contend for bandwidth, so the effective bandwidth available to each is lower.

It is important to understand that actual data transmission is preceded by a series of operations for device and service discovery. Moreover, either a new piconet has to be established or the device has to join a piconet that already includes another peer.

At present, most Bluetooth stacks, including the Nokia Bluetooth stack, do not support scatternets.

4.4.1. Creating a Connection

Since most Bluetooth devices do not support scatternets, they are affected by piconet restrictions. In order to understand how these restrictions affect Bluetooth application development, it is important to understand how the piconet is organized.

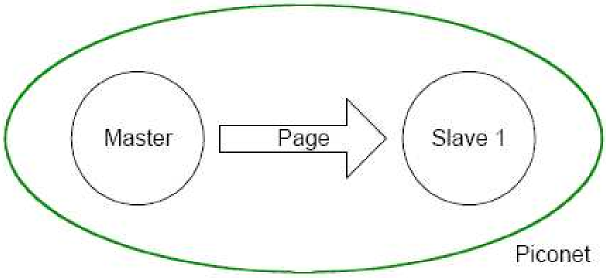


Figure 10: Point-to-Point Connection

The device that initiates creation of a piconet becomes a Host; the device invited to the piconet (device that was paged and is in page scan mode) becomes a Client (see Figure 10). Only the Host can add additional devices to the piconet.

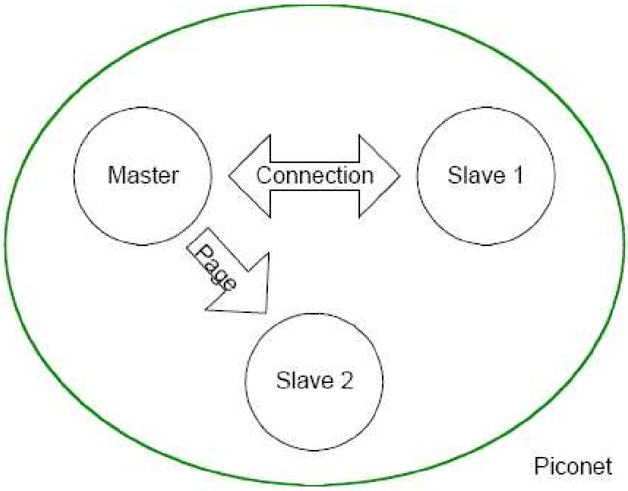


Figure 11: Point-to-Multipoint Connection

If the Host wants to build a point-to-multipoint connection (see Figure 11), it adds the Clients one after another. In the piconet, it is not possible to send data directly from Client 1 to Client 2; data must instead be routed via the Host.

In all other cases, the Bluetooth network turns into a scatternet. If a device does not support scatternet connections, the page procedure simply gets no reply from the addressed device. Thus, the creation of the connection fails.

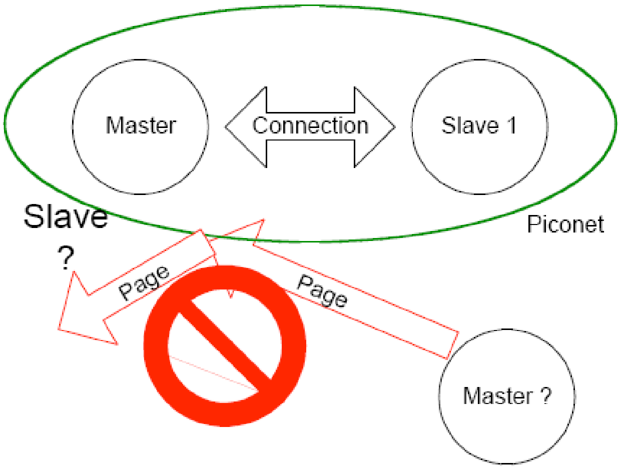


Figure 12: Failed scatternet initiation to Host

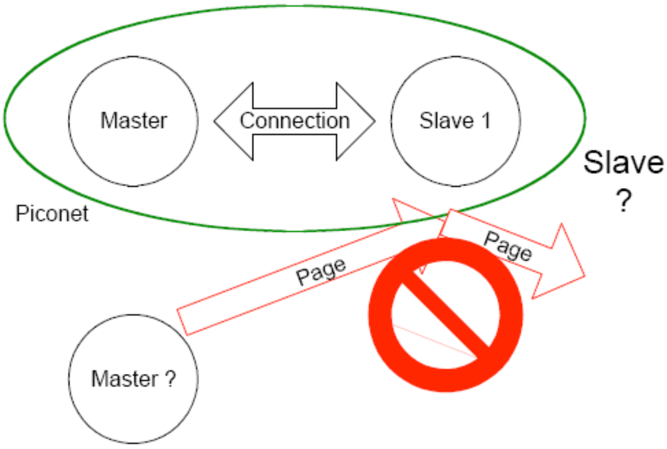


Figure 13: Failed scatternet initiation to Client

In most client/server applications including multi-player games, clients initiate the connection to a server. This will not work in a Bluetooth game because only a Host can initiate a connection, and if multiple devices appoint themselves as Hosts and try to initiate connections, some will fail because scatternets are normally not supported. Therefore, the method shown in Figure 11 is the way to set up a point-to-multipoint connection.

4.5. Bluetooth Protocols

Bluetooth provides a reliable connection; there is no need to add protocol headers for data correction.

Corrupted packets are retransmitted until they are correctly received (even using DH packets). Additional protocols are defined for different purposes.

4.5.1. L2CAP

Logical Link Control and Adaptation Protocol (L2CAP) is the lowest-level Bluetooth protocol that can be accessed by an application. The protocol overhead for L2CAP is 4 bytes. L2CAP is recommended if you have a small amount of data and you need fast response times.

4.5.2. RFCOMM

RFCOMM is a Bluetooth protocol based on L2CAP. The protocol overhead for RFCOMM is between 4 and 5 bytes for small packets. For every 127 bytes of data, the header increases in size by 1 byte. So the overall protocol overhead is about 8 to 9 bytes for data less than 127 bytes (4 bytes from L2CAP and 4 to 5 bytes from RFCOMM). 3 Bluetooth Application Recommendations with Real Time Constraints.

5. Play by Bluetooth

5.1. Games Once Started

Bluetooth provides a reliable connection; there is no need to add a custom protocol for data correction or data acknowledgement. Corrupted packets are retransmitted until they are correctly received.

. Base the data exchange on a protocol that fits your application. Thus, if reducing latency is highly important (as it would be in a fast-action game, for instance) use a protocol with little overhead, such as L2CAP, and try to reduce the data that must be exchanged during game playing to small packets.

5.2. Game Data Examples

There exist a number of different game models as follows:

- **Frame-based:** Each time a device composes a new screen for display on the device, it also performs data exchange. This is only feasible when latency is less than 40 milliseconds (25 frames per second).

- **Dead reckoning:** Data exchange and frame update are done separately. If data is missed, a client uses predictive algorithms to compose the screen, correcting it in future frames as necessary when correct data is received. This scheme works well for Internet-level latencies (100–200 ms for round trip of data). Used in first-person shooter games.

- **Synchronous simulation:** Data exchange and frame update are done separately. Every client has a complete model of the scene. If one client's data is delayed, the Host updates the other clients when the delayed client finally does send update information to the Host. This scheme works well for latencies of up to several hundred milliseconds. Used in massively multi-player online games.

- **Turn-based:** The Host waits for the client data and updates all clients after that data is received; used, for example, in chess. Since players are likely to respond more slowly than the network, higher levels of latency are tolerable— several seconds is fine. Since the frame-based model has the most stringent latency requirement that is why chose this model for our case.

This example was selected to measure the round-trip time in conjunction with Bluetooth and RFCOMM protocol. This is a theoretical example, not a practical one, and was chosen to measure Bluetooth round-trip times.

Host: Transmits data to every client and collects data from all clients. When a client sends data to the host, it sends one packet. When the host transmits to clients, the packet it sends consists of the data received from the clients plus the host's data. If each client sends N number of bytes, the host transmits ([number_of_players] * N) bytes of data to each client.

Client: Receives data from the host, then sends its own data to the host.

Data size: Data shall be exchanged in a single Bluetooth packet. If you need to transmit more data, switch from Dx1 to Dx3 or Dx5 — that is, increase the packet size. Since Bluetooth latency is a function of packet size, doing this multiplies the response and cycle time.

This differs from the typical behaviour of many PC-based games. A game server receives data only from a subset of the clients during one update cycle, because the clients send typically only the changes the user has made — that is, if the keyboard/joystick position has not changed since the last cycle, there's no need to transmit anything. The game state is nevertheless transmitted to all the clients. In other words, this is a worst-case example, since all the clients are transmitting something. The advantage is that no prediction is necessary.

5.3.1. Description for two players

Figure 14 shows the example scenario with one Host (player 1) and one Client (player 2). This scenario differs from many PC-based games. Normally, the frame rate and the game update rate are not directly bound to each other. (The "game update rate" is defined as the number of update messages exchanged between servers per second.) That is, the data throughput of the system doesn't determine how many frames per second the game can have. Typically, there are multiple frames between updates. The game object positions are extrapolated between the frames based on the information that was received in the most recent update.

If it is possible to have a data update every frame, extrapolations are not necessary. This reduces the CPU load for the game, but does mean that if there is some network problem, the game freezes. In practice, it is usually better to perform screen refreshes and data exchange asynchronously.

The figure assumes a two-player game — that is, there is one host and one client. Each has two layers:

1. The application layer
2. The Bluetooth layer

When the game is in play, the following happens on the host side:

- Application layer:

  - The host reads its own game data and receives the client game data from cycle n.
  - The host sends its own plus the client data from cycle n to the client.
  - The host renders the cycle n animation frame based on the data it received from the client and its own data.
  - Repeat for cycle n+1.

- Bluetooth layer:

  - The Bluetooth layer transmits all scheduled packages to the specified client and sends an additional message to the local device when the client has correctly received the package.
  - The Bluetooth layer schedules all received messages to the next layer in the local device.

When the game is in play, this is what happens on the client side:

- Application layer:

  - The client receives its own and the host game data from cycle n.
  - It transmits the game data for the next cycle (n+1).
  - The client renders the cycle n animation frame based on the data received from the host.
  - Repeat for cycle n+1.

- Bluetooth layer:

  - The Bluetooth layer transmits all scheduled packages to the host and sends an additional message to the local device when the package has been correctly received on the host side.
  - The Bluetooth layer schedules all received messages to the next layer in the local device.

BT: *Bluetooth*

Mi: *Host Joystick data of cycle i*

Si: *Clients Joystick data of cycle i*

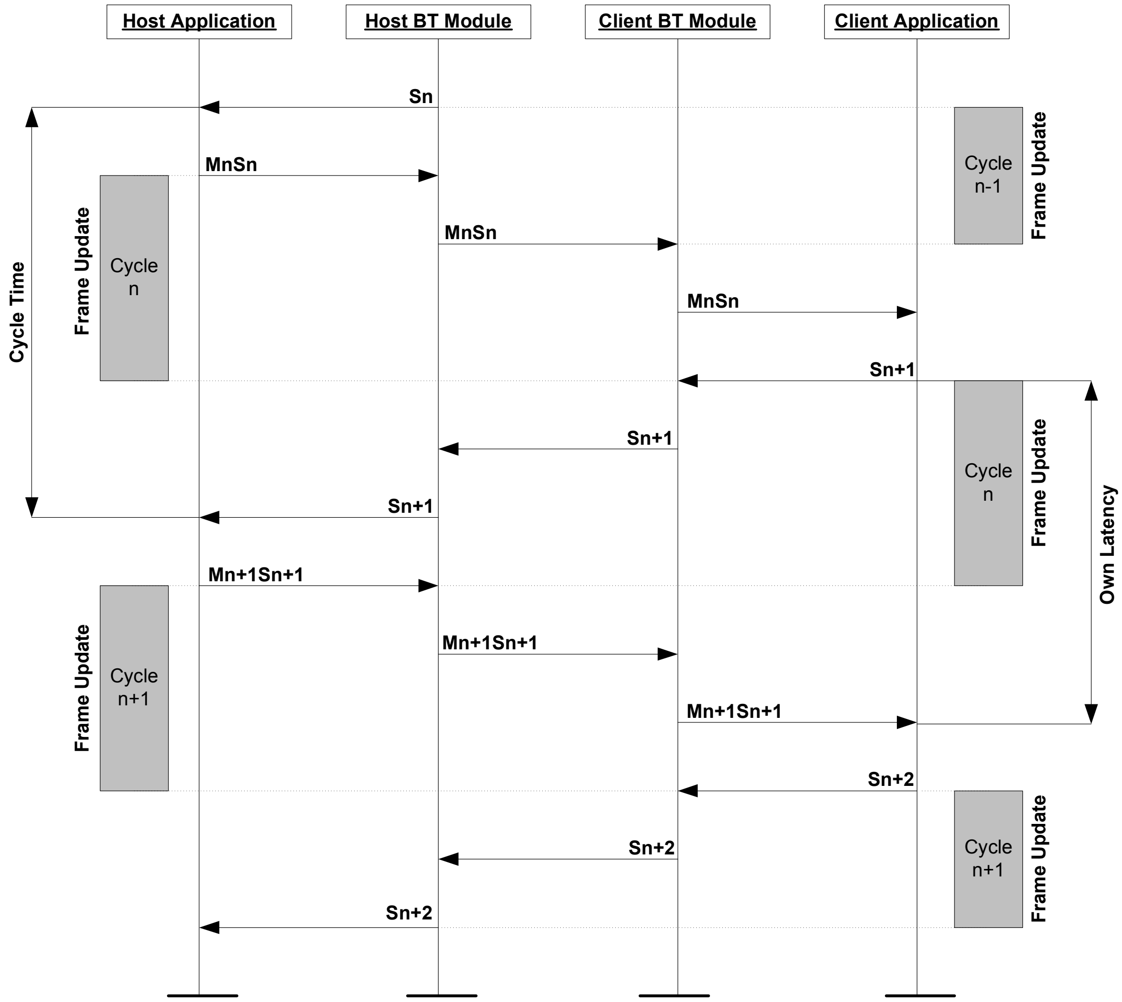


Figure 14: Data exchange sequence chat

As we can see in the Figure 14, the host starts sending *MnSn* to the client and the host waits for the client data and updates all clients after that data is received. Since players are likely to respond more slowly than the network, higher levels of latency are tolerable (several seconds is fine).

5.4. Game Shutdown

When the game is finished, it is important to:

- Set the scan settings back to the values before the game has been started, and
- . Deregister the services in the SDP database.

5.5. Low-Power Mode

Bluetooth modules that reduce power consumption can affect game latency if developers are not careful. After a period in which no Bluetooth data is received, the device enters a low-power mode called SNIFF mode. It checks for data transmission at a longer interval, powering back up when data is received.

In Nokia devices, the device enters SNIFF mode after 15 seconds of no data, and then only checks for data every 0.5 seconds. Thus, the first data exchange will suffer up to an additional 0.5 second delay.

5.6. Disconnection and Link Loss

The game should gracefully handle an unexpected link loss or a disconnection by a user. A link loss will occur when a device does not receive any packet1 from the remote device for more than 20 seconds.

Players do drop out at times, and given the nature of the networking environment, an accidental loss is also possible.

A link loss on the Symbian OS and Java application side is only detected by a failing read or writes operation on Bluetooth. There is no immediate way, when this occurs, to distinguish between a link loss and a purposeful disconnection.

One approach is to have the game application send a message to the host when a player purposefully quits or leaves the game, and assume that in the absence of such a message, a link loss has occurred.

6. Software Analysis

6.1. *Introduction*

This chapter is focused on the software analysis and I have chosen multi-player card game as an example, considering that it is a quite simple game to analyse. The aim is to study the role of each actor and the relation between them.

The actors that are going to take part in this case are the following ones:

- The player/user.
- The mobile phone used as a Client.
- The big screen used as a Host.

Imagine, as an example, a person in a café or bar; the screen (used at other times for TV or newsfeed) switches to game mode and players connect using Bluetooth. A wide range of games are provided, so that the user can decide the game with which he wants to play.

Having this example in mind, I am going to analyse on the one hand the connexion between the Host and the Client and on the other hand the software related to the card game in order to see how the different actors interact.

6.2. *Host – Client Roles*

All Bluetooth activities consume bandwidth, which leads to higher latency for the game. Therefore (depending on the latency requirements of the game), all Bluetooth activities that do not belong to the game should be cancelled (for instance, a currently running Bluetooth audio link or a background file transfer). If the game application is unable to end the background activity, it should provide an alert to the user, asking the user to end or finish the activity before playing.

Then the user should be asked to select a game client or game host role. In a client/server architecture, the client uses a service, which is provided by the server. The host provides the game service to the clients. Because clients want to retrieve information from a server, they normally actively connect to the server. Thus, very often the term “client” is used for the device that initiates a connection. But in this work the terms “client” and “host” are related to the role in the game. Since with Bluetooth only the Host can connect the Clients, the Host is inherently the server, and the Client devices inherently clients.

6.3. Host Role

Typically, in small-scale multi-player games, all devices run the same software, and any can act as host. Developers must take care to ensure that devices do not contend for the role of host, which would cause connection failures, as described previously. One way to do this is to allow any player to tell his or her software, “I wish to serve as host; issue invitations to other devices nearby.” This is more or less how Quake works; any player can serve as host, with other players searching for and connecting to the first player’s machine.

A host should perform the following actions:

1. Disable page scan and inquiry scan.
2. Register the service, which is unique for the game.
3. Start inquiry to obtain all Bluetooth devices with the appropriate CoD field in range. (The host/clients should perform an inquiry/inquiry scan on DIACs[1] to reduce the number of responses.)
4. Perform a service search to find the devices that have started the game and to retrieve the player’s name.
5. Present the found devices that have the game running to the host user.
6. Allow the host user to select multiple devices he or she wants to connect to.
7. Connect the devices the host has selected.

6.4. Client Role

The client should perform the following actions:

1. Register the service, which is unique for the game. Also create a player name as a service attribute. As default, use the Bluetooth device name.
2. Enable page scan and inquiry scan. (The host/clients should perform an inquiry/inquiry scan on DIACs to reduce the number of responses.)
3. Wait for incoming connection request.
4. Present to the user the choice to join or cancel the incoming connection request.
5. Set inquiry scan address back to GIAC[2].

[1] Dedicated Inquiry Access Code

[2] General/Unlimited Inquiry Access Code

6.5. Client and Host Role

The role of the client and host is to exchange necessary information for the game in a setup phase. The total number of players, data size of exchanged packets, and position of client data in the data packet can be negotiated. Once this is done, data needed to support game playing (such as joystick/position information) can be kept small and sent using small and speedy packets.

6.6. Scenarios

One possible definition of scenarios is like a type of work item, describing a specific usage of the envisaged software system by a particular persona. Scenarios should be goal directed. As a persona attempts to reach a goal, the scenario records the specific steps taken in attempting to reach that goal.

We can represent scenarios by writing, Storyboards, video or use case diagram of UML:

- **Writing**: Descriptions by writing are the fastest representation but the principal problem is that it produces ambiguous interpretations of the model or descriptions too longs.
- **Storyboards**: This technique is used for describing concrete situation of scenarios, it will be helpful if you combine with others representations tools, for example writing.
- **Video**: This is the best representation tool, but it is the most expensive because it requires specialised person and equipment.
- **Use case diagram of UML**: In the field of software engineering, the Unified Modelling Language (UML) is a standardized specification language for object modelling. UML is a general-purpose modelling language that includes a graphical notation used to create an abstract model of a system, referred to as a UML model. UML is officially defined at the Object Management Group (OMG) by the UML metamodel, a Meta-Object Facility metamodel (MOF). Like other MOF-based specifications, the UML metamodel and UML models may be serialized in XMI. UML was designed to specify, visualize, construct, and document software-intensive systems.

As we can see in figure 15, the study based on the multi-player card game mentioned in previous chapters, shows on the one hand use cases between users and mobile phones and on the other hand use cases between mobile phones and screens.

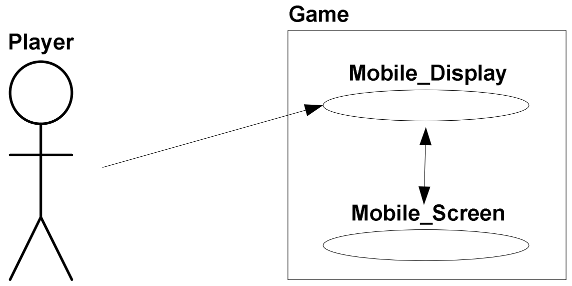


Figure 15: Actors

a. Uses cases of Mobile_Display

1. ***Use case name:*** Connection
2. ***Actors:*** Client, Host
3. ***Function:*** To establish a connection between Host and Client.
4. ***Description:*** Client sends a connection petition to Host
5. ***Tasks:***
   - Client sends to Host a connection petition.
   - Host executes inquiry, service discovery, UUID, remote name request, Host Client roles and security.
   - Host replies to Client with a message accept or not accept, if it is connection possible.
   - Host sends a list of games actives. (In our case it is a cards game).

1. ***Use case name:*** Select_Game
2. ***Actors:*** Client, Host
3. ***Function:*** To select the game.
4. ***Description:*** Client sends to Host the game selected and Host responses Client with message “OK” if it is possible.
5. ***Tasks:***
   - Client sends to Host the game selected.
   - Host replies to Client with a message “OK” and runs the game or “This game is unavailable, try another one”.

1. ***Use case name:*** New_Game
2. ***Actors:*** Client, Host
3. ***Function:*** To start a game.
4. ***Description:*** Host and Client start to sends and receive game dates.
5. ***Tasks:***
   - Client sends and receives game dates.
   - Host sends and receives game dates.

1. ***Use case name:*** Game_Details
2. ***Actors:*** Client, Host
3. ***Function:*** To give details.
4. ***Description:*** Host sends to Client game details.
5. ***Tasks:***
    - Host sends to Client the shift.
    - Host sends to Client the number of plays.
    - Host sends to Client the number of points accumulates.

1. ***Use case name:*** End_Game
2. ***Actors:*** Client, Host
3. ***Function:*** To finish the game.
4. ***Description:*** Client sends to Host a end game petition and Host accepts.
5. ***Tasks:***
    - Client sends a end game petition.
    - Host receive the petition and close the game and sends a accept message.

1. ***Use case name:*** End_Connection
2. ***Actors:*** Client, Host
3. ***Function:*** To finish the connection.
4. ***Description:*** Client sends to Host a disconnect petition and Host accepts.
5. ***Tasks:***
    - Client sends disconnect petition.
    - Host receive the petition and sends a accept message and disconnect the device.

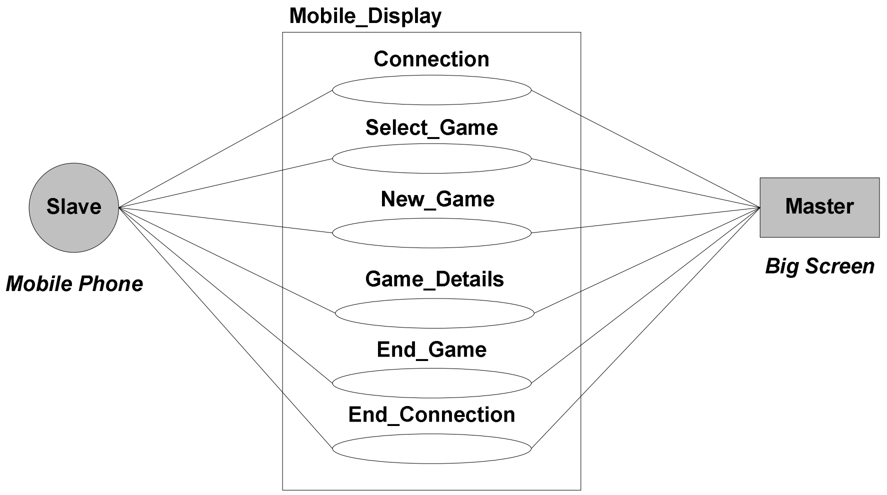


Figure 16: Interaction between Host and Client

b. Uses cases of Mobile_Screen

1. ***Use case name:*** Open_Bluetooth
2. ***Actors:*** Player, Client
3. ***Function:*** To activate the Bluetooth.
4. ***Description:*** The player selects in the Bluetooth panel the option to activate Bluetooth.
5. ***Tasks:***
   - Player selects to open Bluetooth panel.
   - Player selects activate Bluetooth.

1. ***Use case name:*** Search_Devices
2. ***Actors:*** Player, Client
3. ***Function:*** To find Bluetooth devices.
4. ***Description:*** The player selects in the Bluetooth panel to search devices.
5. ***Tasks:***
   - Player selects to open Bluetooth panel.
   - Player selects search devices.

1. ***Use case name:*** Connection
2. ***Actors:*** Player, Client
3. ***Function:*** To start connection.
4. ***Description:*** The player selects the correct device and starts the connection.
5. ***Tasks:***
   - Player selects the device.

- Player selects start connection.

1. ***Use case name:*** Select_Game
2. ***Actors:*** Player, Client
3. ***Function:*** To select the Game.
4. ***Description:*** The player selects the game in the mobile screen.
5. ***Tasks:***
    - Player selects the game.

1. ***Use case name:*** New_Game
2. ***Actors:*** Player, Client
3. ***Function:*** To start a Game.
4. ***Description:*** The player selects the option in the mobile screen to start the game.
5. ***Tasks:***
    - Player starts the game.

1. ***Use case name:*** Select_Card
2. ***Actors:*** Player, Client
3. ***Function:*** To chose a Game.
4. ***Description:*** The player selects a card in the mobile screen.
5. ***Tasks:***
    - Player chooses a card.

1. ***Use case name:*** Leave_Card
2. ***Actors:*** Player, Client
3. ***Function:*** To leave a card.
4. ***Description:*** The player selects the card in the phone screen that he wants to leave in the big screen.
5. ***Tasks:***
    - Player leaves a card.

1. ***Use case name:*** Pick_up_Card
2. ***Actors:*** Player, Client
3. ***Function:*** To pick up a card.
4. ***Description:*** The player will select the option pick up cards, when he wants to go next turn.
5. ***Tasks:***
    - Player selects to pick up cards.

1. ***Use case name:*** End_Game
2. ***Actors:*** Player, Client
3. ***Function:*** To finish the game.
4. ***Description:*** Player selects in the mobile phone end game.
5. ***Tasks:***
    - Player selects end game.

1. ***Use case name:*** End_Connection
2. ***Actors:*** Player, Client
3. ***Function:*** To finish the connection.
4. ***Description:*** Client selects in the mobile phone to end the game.
5. ***Tasks:***
    - Player selects finish connection.

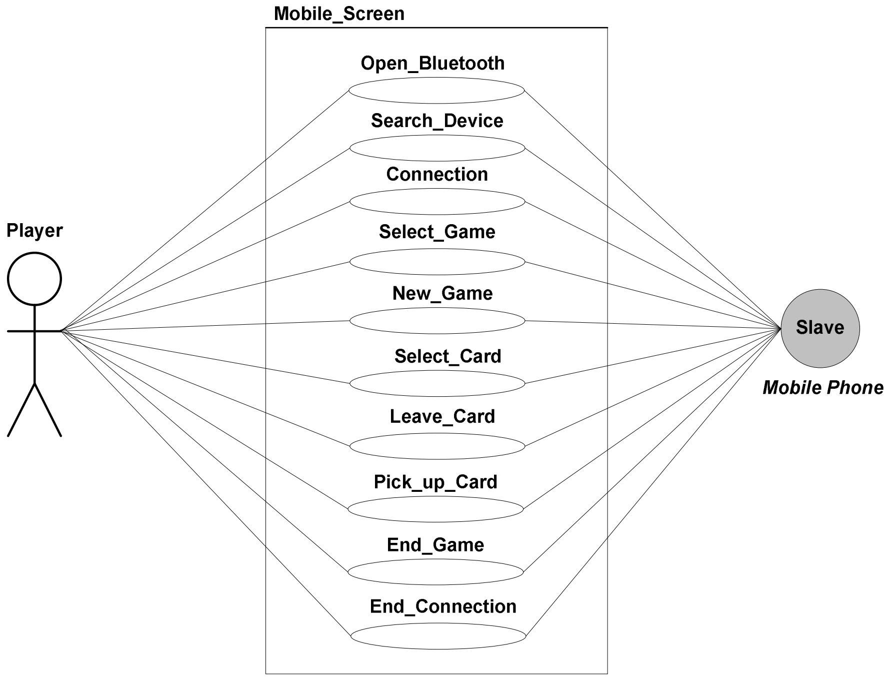


Figure 17: Interaction between Player and Client

Finally, this next figure shows a process diagram of the multi-player card game where it is clearly shown the relation between the three actors (the player, the mobile phone and the display) and processes executed by them.

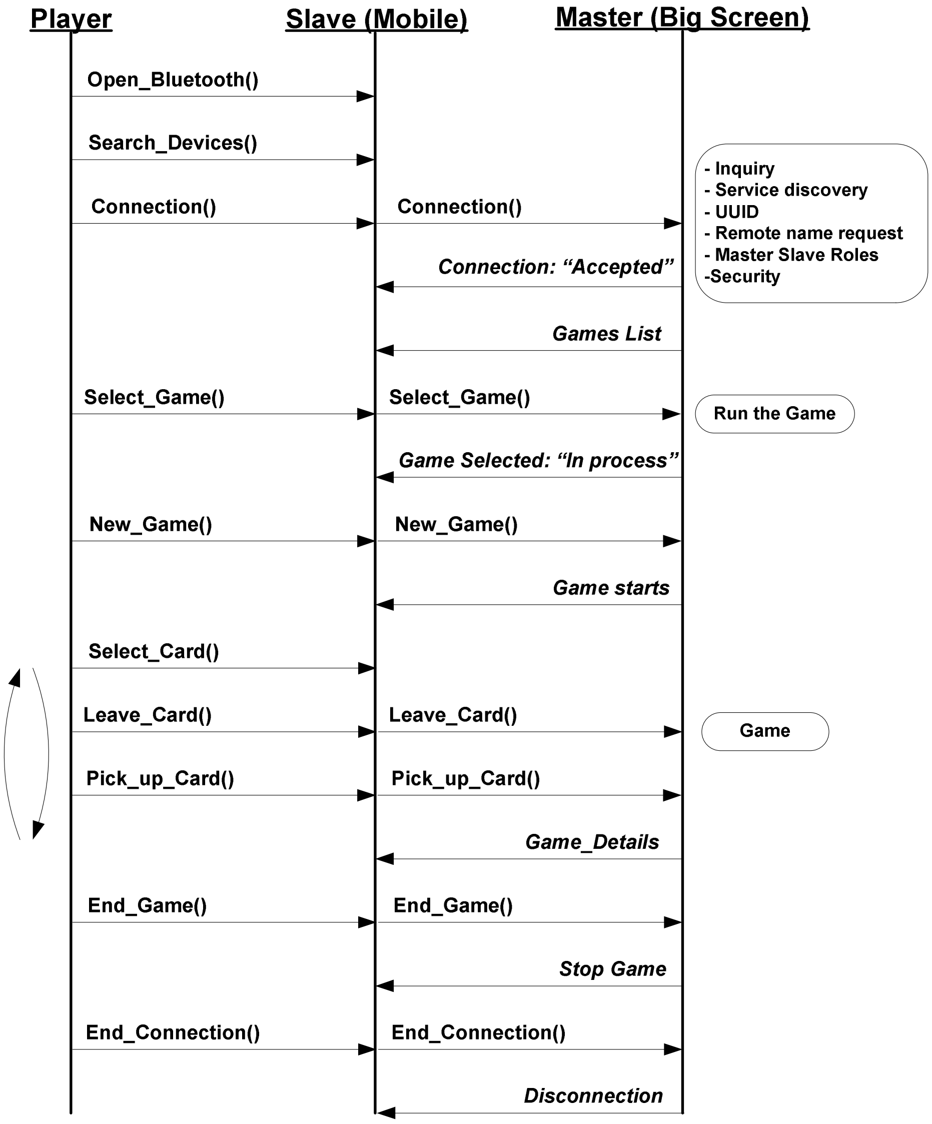


Figure 18: Process diagram of cards game

7. Heuristic Evaluation

7.1. Introduction

The evaluation stage is a key point in order to obtain usable and accessible interactive systems. It is in this stage where the necessary techniques are going to be employed to receive values by users and/or expert evaluators and where interactive procedures are going to be improved.

An evaluation is an activity that comprises a group of methodologies and techniques which analyse the usability and accessibility of an interactive system in different stages of the software life cycle.

The aim of the evaluation is the identification of particular problems related to the design and which could cause unexpected results or confusion between different users.

The evaluation method used is based on the inspection technique. The term inspection applied to the usability comprises a group of methods to evaluate the usability. The evaluators explain the usability grade of a system on the base of the inspection or the interface test.

In the inspection evaluation there are different methods followed underlining the Heuristic Evaluation as one of the most important.

A heuristic evaluation is a guideline or general principle or rule of thumbs that can guide a design decision or be used to critique a decision that has already been made. Heuristic evaluation, developed by Jakob Nielsen and Rolf Molich, is a method for structuring the critique of a system using a set of relatively simple and general heuristics. Heuristics evaluations can be performed on prototypes, storyboards and fully functioning systems. It is therefore a flexible, relatively cheap approach. Hence it is often considered a discount usability technique.

For our case I will used the Heuristic Evaluation methods for Mobile Computing. This evaluation describes a modified collection of usability heuristics that are designed to be appropriate for evaluation in mobile computing.

7.2. Nielsen Heuristic Evaluation

The general idea behind heuristic evaluation is that several evaluators independently critique a system to come up with potential usability problems. It is important that there be several of these evaluators and that the evaluations be done independently. Nielsen's experience indicate that between three and five evaluators is sufficient, with five usually resulting in about 75% of the overall usability problems being discovered.

To aid the evaluators in discovering usability problems, a set of 10 heuristics are provided:

1. **Visibility of system status:** The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

2. **Match between system and the real world:** The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

3. **User control and freedom:** Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4. **Consistency and standards:** Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

5. **Error prevention:** Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

6. **Recognition rather than recall:** Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7. **Flexibility and efficiency of use:** Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8. **Aesthetic and minimalist design:** Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9. **Help users recognize,** diagnose, and recover from errors: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. **Help and documentation:** Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

7.2.1. Strengths of Heuristic Evaluation

Here are some strengths of heuristic evaluation:

- Its ease of implementation and high efficiency. It is considered to have a good success rate in that typically only 3–5 usability experts are needed to detect most (75–80%) of the usability flaws a system presents.
- Its early applicability in the development lifecycle and low cost: it requires neither a working prototype nor the real users.
- It is becoming part of the standard HCI curriculum and therefore known to many HCI practitioners. The heuristics are well documented and therefore easy to learn and put to use, so it may be argued that heuristic evaluation can also be effectively conducted by non-usability experts.

On the whole, heuristic evaluation is considered to be a cost-effective evaluation method. Its main strengths lie in providing discovery and analysis resources, such as domain and system knowledge, where it generally outperforms other popular inspection techniques like guideline-based methods or cognitive walkthrough.

7.2.2. Limitations of Heuristic Evaluation

Here are some specific limitations of heuristic evaluation:

- Heuristic evaluation is highly dependent on the skills and experience of the specific usability expert(s) involved. At a high level of generality, the heuristics are “motherhood statements that serve only to guide the inspection rather than prescribe it” (Greenberg et al., 1999).

- Participants are not the real users. Regardless of the experts' skills and experience, they are still “surrogate users” (i.e. experts who emulate real users), therefore the resulting data are not really representative of the real users.
- Heuristic evaluation does not fully capture or take into account the context of use of the system under evaluation but rather evaluates it “as a relatively self-contained object” (Muller et al., 1995).
- It has been said that the majority of usability flaws detected by heuristic evaluation are ‘minor’ usability problems, or false positives, problems that do not negatively impact user performance or users' perception of system quality (Simeral and Russell, 1997).

When compared to other expert techniques, such as guideline-based methods and cognitive walkthrough, heuristic evaluation is strong in terms of thoroughness (percentage of problems found), but weak in terms of efficiency (number of true positives vs. false positives) and, like other inspection methods, is vulnerable to expert biases.

7.3. Mobile Usability Heuristics

Mobile computing devices are typically smart products or information appliances, and are generally consumer products. Their users are thus a heterogeneous group and so it may be more difficult to find suitable surrogate users for user-based testing in mobile computing.

I believe that heuristic evaluation can be enriched and adapted toward capturing contextual factors. Our focus is on usability problems in mobile devices and a discussion of their sources. However, this should be set against a broader view of the fantastic world of new opportunities, advantages, and benefits that mobile devices and contexts bring. While we will not touch explicitly on this again, the chapter should be read in the light that the problems and limitations are ones worth tackling because of the opportunities offered by the technology.

This evaluation is based on a document, mentioned at the bibliography, and which was carried out by Alan Dix.

1. Visibility of system status and losability/findability of the mobile device: Through the mobile device, the system should always keep users informed about what is going on. Moreover, the system should prioritize messages regarding critical and contextual information such as battery status, network status, environmental conditions, etc. Since mobile devices often get lost, adequate measures such as encryption of data should be taken to minimize loss. If the device is misplaced, the device, system or application should make it easy to recover it.

2. Match between system and the real world: Enable the mobile user to interpret the information provided correctly, by making it appear in a natural and logical order; whenever possible, the system should have the capability to sense its environment and adapt the presentation of information accordingly.

3. Consistency and mapping: The user's conceptual model of the possible function/interaction with the mobile device or system should be consistent with the context. It is especially crucial that there be a consistent mapping between user actions/interactions (on the device buttons and controls) and the corresponding real tasks (e.g. navigation in the real world).

4. Good ergonomics and minimalist design: Mobile devices should be easy and comfortable to hold/carry along as well as robust to damage (from environmental agents). Also, since screen real estate is a scarce resource, use it with parsimony. Dialogues should not contain information that is irrelevant or rarely needed.

5. Ease of input, screen readability and glancability: Mobile systems should provide easy ways to input data, possibly reducing or avoiding the need for the user to use both hands. Screen content should be easy to read and navigate through notwithstanding different light conditions. Ideally, the mobile user should be able to quickly get the crucial information from the system by glancing at it.

6. Flexibility, efficiency of use and personalization: Allow mobile users to tailor/personalize frequent actions, as well as to dynamically configure the system according to contextual needs. Whenever possible, the system should support and suggest system-based customization if such would be crucial or beneficial.

7. Aesthetic, privacy and social conventions: Take aesthetic and emotional aspects of the mobile device and system use into account. Make sure that users' data is kept private and safe. Mobile interaction with the system should be comfortable and respectful of social conventions.

8. Realistic error management: Shield mobile users from errors. When an error occurs, help users to recognize, to diagnose, if possible to recover from the error. Mobile computing error messages should be plain and precise. Constructively suggest a solution (which could also include hints, appropriate FAQs, etc). If there is no solution to the error or if the error would have negligible effect, enable the user to gracefully cope with the error.

8. Evaluation

8.1. Introduction

On the previous chapter we have already seen the Heuristic evaluation which leads us to the following evaluation. This paragraph is going to be focused on the mobile phone use as an execution game device. The mobile phone brand used in particular cases is the NOKIA 3250[1] taking as a reference the Play Station Portable[2] (PSP) and the card game mentioned in previous chapters.

8.2. Punctuation Method

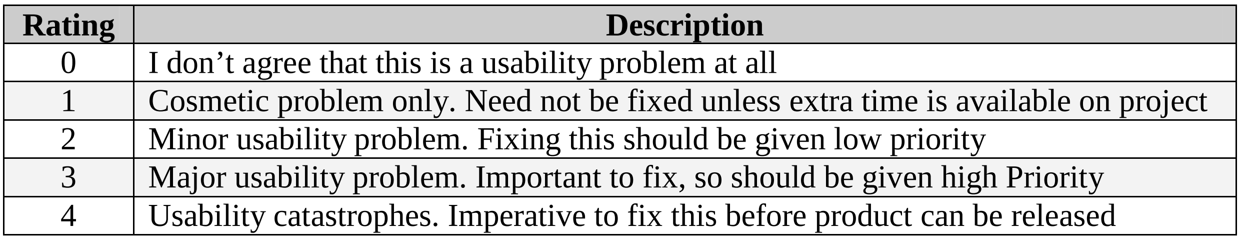
The punctuation method is the following one:

- Each characteristic of each heuristic rule has been analysed and assessed according to the following scale:

  Grade of satisfaction:

  0. Insufficient
  1. Sufficient
  2. Excellent

- After having analysed and assessed the level of satisfaction of the characteristics of the heuristic rule, a global evaluation according to the Nielsen's five-point Severity Ranking Scale (SRS) (Nielsen, 1994b) it is described as follows:



Rating	Description
0	I don’t agree that this is a usability problem at all
1	Cosmetic problem only. Need not be fixed unless extra time is available on project
2	Minor usability problem. Fixing this should be given low priority
3	Major usability problem. Important to fix, so should be given high Priority
4	Usability catastrophes. Imperative to fix this before product can be released

Figure 19: Severity Ranking Scale (SRS)

[1] See Appendix A
[2] See Appendix B

8.3. Evaluation

1. Visibility of system status and losability/findability of the mobile device

In this heuristic we are going to see the facts that keep users informed about what is going on as well as adequate measures such as encryption of data and how the system or application recovers the device in case it is misplaced.

- Which information gives the mobile phone?

In the figure 20 we can see the main screen of the Operating System Symbian S60.

- Battery Status
- Connectivity Signal
- Bluetooth Signal
- Date
- Clock

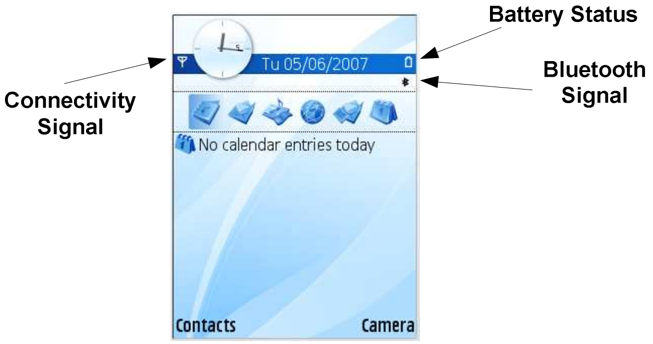
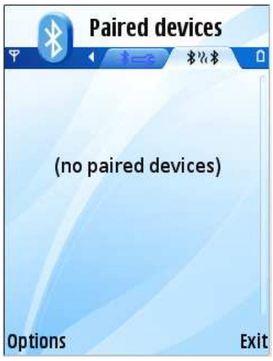


Figure 20: Main screen of the Operating System Symbian S60

The user is informed in different aspects such as the connectivity status, the battery status and whether he/she is connected or not to the Bluetooth device. In addition, it is necessary to underline the fact that this system has a sonorous device which is activated when the level of battery is extremely low. Therefore, all these measures are not only keeping the user informed of the status of his or her mobile phone but also have excellent results in the battery device.

In addition to this, devices have different types of features which keep the user informed.
The figure 21 shows on the left-hand side the number of detected Bluetooth devices, in this case the access would be through *connectivity>Bluetooth>Paired devices.* On the right–hand side we can see the menu that provides settings in order to revise the general status of the mobile phone, in this particular case the access would be through the main menu.


Paired devices

(no paired devices)

Figure 21: Paired devices

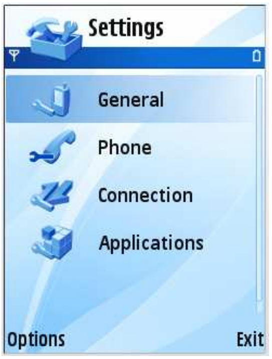

Settings

Figure 22: Settings menu

The results are not satisfactory; neither in the measures of encryption of data or in the system recovery, due to the fact that the only security measure available in this system is an access code to the SIM card and to the memory card. We have to underline the fact that there is not any security system in case of a loss of the mobile phone which means that the mobile phone itself does not provide any recovery system.

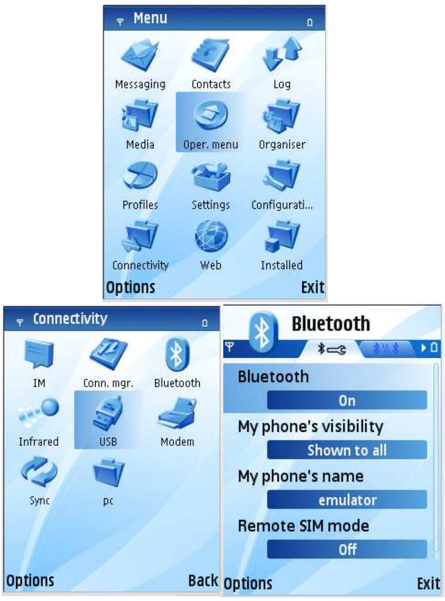
	Satisfaction
Battery status	2
Connectivity signal	1
Bluetooth signal	1
List of Bluetooth Devices detected	1
Adequate measures such as encryption of data	0
Grade of recovery	0
Severity Ranking Scale: 1	

As the Battery status, the connectivity signal, the Bluetooth signal and the list of Bluetooth devices detected are satisfactory and are more important than the adequate measures and the grade of recovery the qualification of the SRS is 1.

2. Match between system and the real world

The following heuristic evaluates whether the information provided by the mobile phone is correct, accurate and useful and whether it appears in a natural and logical order.

Figure 23 and 24 shows different types of direct accesses and how information is organized.



Bluetooth

Bluetooth
On

My phone's visibility
Shown to all

My phone's name
emulator

Remote SIM mode
Off

Figure 23: Main menu, Connectivity menu, Bluetooth menu

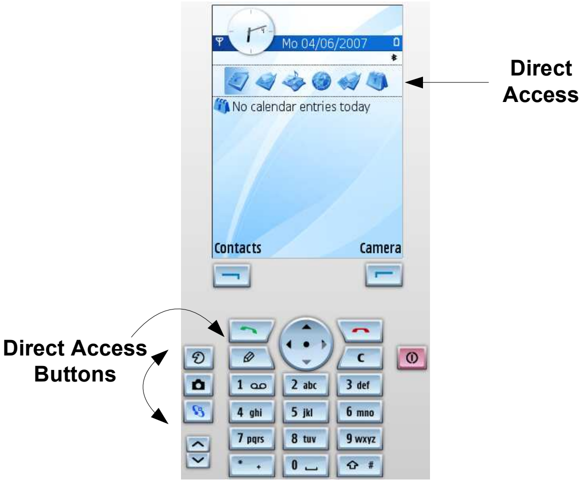


Figure 24: Emulator of NOKIA with Symbian S60 OS

The figure shows three types of menus: main menu, connectivity menu and Bluetooth.

First of all, it has to be underlined the fact that there are some utilities of the main menu which are universally and frequently used such as the messaging, contacts and dialled options and secondly, the fact that the connectivity menu and the Bluetooth are directly connected and you need both to establish certain connections.

In these cases, functionalities have a glyph used in the visual representation of characters, such as the messaging option which is represented by an envelope or the USB connection which is represented by a slot of the connection. That differs from the Bluetooth menu in which information is not represented by glyphs but by short sentences which describe accurately all functionalities.

On the right hand side a representation of a Symbian S60 emulator that offers different direct accesses which allow the user to have a quicker connection to certain options such as the messaging, contact or digital camera.

This heuristic has shown that users have direct access to some of the options seen previously which not only give correct and accurate information, but also make easy the usability and accessibility of the mobile phone. Nevertheless, it has to be considered the fact that direct accesses in the case of game applications do not exist because they are still not considered regular actions. This is the reason why the level of satisfaction of some accesses such as the Bluetooth panel and the game execution, shown on the next table, are not excellent.

	Satisfaction
General Representation	2
Bluetooth panel	1
Game execution	1
Severity Ranking Scale: 2	

In conclusion, the evaluation of this heuristic is satisfactory for the reasons explained in the last paragraph. The final qualification of the SRS is 2.

3. Consistency and mapping

This section evolves the user's conceptual model of function/interaction with the mobile phone device or system and its consistency.

In this heuristic I am going to assess the consistency of the mobile device or system with the context as well as the consistent mapping between user actions/interactions and the corresponding real task. In order to carry out this analysis I am going to use some of the aspects already analysed in previous chapters such as chapter II in which I mentioned the situational context, such as location, surrounding environment or state of the device, as implicit input to the system and chapter III in which I defined the limits posed by the device and the limits posed by context and interaction.

After having done this introduction I am going to evaluate each of the following concepts according to the case I am analysing:

- **Limits posed by the device**
  - *Small-screen*
  - *Limited input*
  - *Limited bandwidth and cost*
  - *Limited connectivity*
  - *Limited computational resources*
  - *Limited power*
  - *Wide heterogeneity*

- **Limits posed by context and interaction**
  - *Variable context*
  - *Kind of interaction*
  - *Interruptions*
  - *Privacy and security*
  - *Intimacy and availability*

Limits posed by the device refer to all the technical characteristics limiting mobile phone devices.

In the case I am analysing, a design of a multi-player game, it is important to consider the limitations I have mentioned before because they determine whether it is possible to execute certain games or not.

In order to better understand what I have mentioned I am going to analyse first of all the limits posed by the device and secondly the limits posed by context and interaction.

1. **Small – screen:** Most of the mobile phones, such as NOKIA 3250[1] (**176 x 208 pixel** TFT active matrix 262,144-color display i size 34.8mm x 41.1mm), do not have screens big enough to execute certain games such as fighting games, races or persecution games, although we already have some devices such as the Play Station Portable[2] (TFT panoramic (16:9), with more than 16 millions of colours and the resolution is **480 x 272 pixels),** which are suitable to execute some of the games considered before.

2. **Limited input:** In general, all games need a mini joystick in order to control directions and other buttons for common actions such as jump, run and so on. In this case, there isn't any input limitation which means that with the keyboard provided and with the central device which controls the selection of the menu options is sufficient to carry out any type of game, so that the evaluation for this case results satisfactory.

3. **Limited bandwidth and cost:** mobile phones have access to internet, although it is slow and expensive. This information is irrelevant considering our analysis which means I am not going to evaluate it.

[1] See Appendix A
[2] See Appendix B

4. **Limited connectivity:** In this paragraph I am going to analyse the characteristics of the Bluetooth connexions in order to execute multi-player games.
   As mentioned in chapter IV, the type of connection used is going to be a scatternet of 6 players and a Host with a shared broadband. That makes us going back to the PSP which uses a Bluetooth connection and which shows that it allows the necessary data exchange. In conclusion, the Bluetooth connection results satisfactory.

5. **Limited computational resources:** This case refers to the hardware limitations which determine the development of different kinds of games. Phones, as seen in previous chapters, have hardware to execute different kinds of games, though not of the last generation and which just satisfy the basic needs. In conclusion, the hardware is satisfactory.

6. **Limited power:** Nokia 3250 and most of the mobile phones have an autonomy of three hours with talk time, in this case I am also going to be refereeing to the PSP device which has an autonomy of 6 hours when it has to execute some of the games. In conclusion, the level of autonomy is also satisfactory.

7. **Wide heterogeneity:** this case analyses the wide of heterogeneity mobile phones need to be adapted to all types of devices. Considering this research which is based on the Symbian operative system (version: S60), games should be adapted to all different kinds of screens. It has to be considered that nowadays all games take this premise into account which means that the heterogeneity is satisfactory.

Limits posed by context and interaction

1. **Variable Context:** Mobile phone devices are designed to be used in all contexts. The case I am analysing (a multi game, in a bar or pub) would not be an exception. In conclusion, this variable context is excellent.

2. **Kind of interaction:** the connection and execution of the game should be easy and quick. The device I am working with satisfies the interaction, although considering that it does not have any direct access it is not easy and quite slow. In conclusion, this variable is satisfactory.

3. **Interruptions**: this variable refers to all actions taken by the software and which alert the user of any important notice which has to be considered during the execution, such as low battery, messages received, phone calls and so on. In general, interruptions can not be administered by the user who should decide whether to receive a call or a

message during the execution of a game or not. So the conclusion for this variable is not satisfactory.

4. **Privacy and security:** As seen in previous chapters, privacy and security is related to the Bluetooth connection, which is the reason why the evaluation in this case is satisfactory.

5. **Intimacy and availability:** this variable refers to the relation established between the mobile phone and the user. Considering that an option to execute games is added to the mobile phone as a regular user's action the relation between the phone and the user is reinforced, reason why the evaluation is satisfactory.

	Satisfaction
Small- Screen	0
Limited input	2
Limited connectivity	1
Limited computational resources	1
Limited power	1
Wide heterogeneity	1
Variable Context	2
Kind o interaction	1
Interruptions	0
Privacy and security	1
Intimacy and availability	1
Severity Ranking Scale: 3	

In general, the variables are satisfactory, although in some cases, as we have seen through the chapters, they are not, such as in the screen types, the connexion types, the hardware type and the level of power, reason why the qualification of the SRS is 3. It is necessary to make progress in order to have an adequate satisfaction and an adequate SRS level.

4. Good ergonomics and minimalist design

Nowadays, mobile phones have different functionalities which are related to the design. This is the case of a mobile phone designed to be used as an MP3 or used to execute games. Such mobile phones need both image and sound characteristics, reason why some functionalities are going to be affected negatively, such as the size which is considered to be on the one hand optimum for the execution of music or games, but on the other hand extremely big to be used as a mobile phone.

The characteristics of the NOKIA 3250 are the following ones:

1. **Easy to carry on:** this mobile phone allows different functionalities such as the execution of different games, MP3, camera and video. Although does not satisfy the easiness to carry it on.
2. **Mobile size:** the mobile size is acceptable and it satisfies the variable considering that the size of the PSP[1] device is a little bit bigger than the size of the mobile phone (170 mm x 74 mm x 23 mm versus 103.8 x 50 x 19.8 mm).
3. **Buttons distribution:** As seen previously, this type of mobile phone has enough buttons to execute different functionalities, but it has an inconvenient which is the distribution. If we analyse the distribution of the PSP[2] device we notice that both the roll ball and other buttons such as the jumping or shooting ones are located on the same side which means that the distribution of the buttons is not adequate to play and it affects consequently to the execution of some games. So that this variable is not satisfactory.
4. **Press more than 2 buttons:** Most of the games require two options using two different buttons at the same time. Shooting and jumping at the same time could be an example to illustrate this fact. Nowadays as this functionality is not activated this variable is not satisfactory.
5. **Quality of sound:** The sound quality is satisfactory considering that users can regulate the volume as they have their own headphones.

	Satisfaction
Easy to carry on	0
Mobile Size	1
Buttons distribution	0
Press more than 2 buttons	0
Quality of Sound	2
Severity Ranking Scale: 3	

In this case, the qualification of the SRS is 3 considering that three of the variables analysed are unsatisfactory.

5. Ease of input, screen readability and glancability

In this heuristic we will see whether mobile systems should provide easy ways to input data, whether the user should use both hands or not, whether the screen content should be easy to read and navigate through notwithstanding and light conditions.

1. **Easy to input data:** It is really easy to introduce input data such as names or any other kind of information which means that this function is excellent.

[1] See Appendix B

[2] See Appendix B

2. **The need for the user to use both hands:** the use of both hands would be positive because it would offer the user a wide range of different games which can not be offered nowadays, reason why this variable is unsatisfactory.

3. **Screen content should be easy to read/Light conditions:** Nowadays the content of any mobile phone is easy to read because of screen's luminosity and definition. So that this variable is excellent.

	Satisfaction
Easy to input data	2
Need for the user to use both hands	0
Screen content should be easy to read and navigate	2
Light conditions	2
Severity Ranking Scale: 3	

The main inconvenient of mobile phones, as seen previously, is the fact that it is still not possible to use both hands in some of the games which are being designed which means that the final qualification of the SRS is 3.

6. Flexibility, efficiency of use and personalization

Allow mobile users to tailor/personalize frequent actions, as well as to dynamically configure the system according to contextual needs. Whenever possible, the system should support and suggest system-based customization if such would be crucial or beneficial.

In this context any of the personalization actions are excellent which means that it is necessary to incorporate new functionalities.

	Satisfaction
Flexibility	1
Efficiency	1
Personalization	1
Severity Ranking Scale: 2	

In conclusion and in this particular case the qualification of the SRS is 2, taking into account the fact that mobile phones are still not adapted to the context.

7. Aesthetic, privacy and social conventions

Take aesthetic and emotional aspects of the mobile device and system use into account. Make sure that users' data is kept private and safe. Mobile interaction with the system should be comfortable and respectful of social conventions. The sociology a social rule refers to any social convention commonly adhered to in a society.

1. **Aesthetic:** the aesthetic evaluation is satisfactory, considering that it should be more aggressive, modern and adapted to the user needs.
2. **Privacy and social convention:** the privacy and social convention evaluation is excellent, although they both are aspects not particularly taken into account in this case.

	Satisfaction
Aesthetic	1
Privacy	2
Social convention	2
Severity Ranking Scale: 2	

Considering that the aesthetic is the only aspect that has some defects the qualification of the SRS is 2.

8. Realistic error management

In this heuristic I am going to evaluate the error management. Usually errors are caused when users execute an action of an application or a predetermined function and the mobile device can not execute it. If the error is because of the application it has to be considered whether it is because the format is not compatible with the mobile phone or because there is a lack of software. In this case an error message is shown to the user.

If the error is caused because of any other predetermined functionality in the operating system the error message is shown in the failed file.

	Satisfaction
Users to recognize the error	1
Mobile computing error messages should be plain and precise	1
Severity Ranking Scale: 1	

In conclusion, considering that the operating system reports any error occurred letting the user know whether the error is from the system or from any other predetermined function the qualification of the SRS is 1.

8.4. Results

Heuristics	SRS
1. Visibility of system status and losability/findability of the mobile device	1
2. Match between system and the real world	2
3. Consistency and mapping	3
4. Good ergonomics and minimalist design	3
5. Ease of input, screen readability and glancability	3
6. Flexibility, efficiency of use and personalization	2
7. Aesthetic, privacy and social conventions	2
8. Realistic error management	1

Results have been quite positive considering that the heuristics have been evaluated between 1 and 3 and we have not had any 4, which would have meant having usability catastrophes.

Cosmetic problems have been detected in the first and last heuristic, although there is no need to be fixed unless extra time is available on project.

Heuristics 2, 6 and 7 have the following minor usability problems:

- Access to Bluetooth panel and Game execution not direct.
- Personalization: The system has to be configured according to contextual needs.
- Aesthetic should be more aggressive, modern and adapted to the user needs.

Heuristics 3, 4 i 5 have the following major usability problems:

- Small-Screen
- Impossible press more than 2 buttons
- Buttons distribution
- Easy to carry on
- Need for the user to use both hands

Nowadays mobile phones are not designed for the execution of games as a main functionality that is the reason why most of the problems detected are related to the physical and software design. Adding new designs and software adaptations a mobile phone totally adapted to the game execution could be.

9. Conclusions

The main aim of this research, considering the mobile phone use as a game execution, has been to work on the integration and adaptation of computational procedures carrying out a Heuristic Evaluation.

Firstly, the HCI has been defined, as seen in chapter II, explaining both its main goals and its interdisciplinary character. Consequently, the definition has been slightly modified accordingly to the new mobile technology giving more importance to the context and scenario, adding the new Ubiquious Computing paradigm and replacing the computer with the mobile phone.

Secondly, Usability Engineering and Centred User Design have been defined relating concepts to the HCI, as seen in chapter III, determining mobile technology limitations in order to better understand how usability in mobile computing can be evaluated and improved and classifying mobile usability issues which have to be considered in order to design any kind of software.

Thirdly, an enumeration of the mobile phone components has been carried out, as seen in chapter IV. The S60 version of the Symbian Operating System and the Java standard has been introduced in order to programme the mobile technology. After having done an introduction of the Bluetooth connection I have realized that in our case a Point – to – Multipoint Piconet connection was necessary, although considering the fact that the maximum number of customers who could be connected could not be higher than 7.

Furthermore, as seen in chapter V, a theoretical example has been developed following the Turn-based data transfer model to measure the round-trip time in conjunction with Bluetooth and RFCOMM protocol. So that the scenario between both players has been clearly represented and can be used for further researches in order to design certain games.

In addition, on the one hand the host role and the client role has been analysed and on the other hand, an analysis of the software for a multi-player card game has been carried out. As a result each actor has been understandably defined as well as the relation between them, the actions taken by them and the scenario.

Also, in chapter VII, the Heuristic Evaluation of Nielsen and Molich has been introduced as an evaluation method based on the inspection technique, analysing its strong and weak points. Finally, a set of 8 mobile heuristics have been provided in order to put them into practice in the following chapter.

In conclusion, thanks to the evaluation carried out in chapter VIII, mobile phones are not designed for the execution of games as a main functionality that is the reason why most of the problems detected are related to the physical and software design. Adding new designs and software changes a mobile phone totally adapted to the game execution could be configured.

This research has been focused on the results of an inspection method of a Heuristic evaluation which has been carried out by an evaluator. My proposals considering future researches are as follows:

- Continuity of the heuristic evaluation carried out by other evaluators and comparison of results.
- Software design of a multi–player card game.

I would not like to finish this research without stating the fact that this experience has been completely constructive and favourable in all aspects considering that apart from learning English I have worked with Alan Dix who have showed me new facets on this field.

Appendix A

This project has been focused on NOKIA 3250 which uses Symbian S60 Operating System and which was provided by the Lancaster University.

Nokia 3250 features at a glance:

Figure 25: NOKIA 3250

- Dimensions: 103.8 x 50 x 19.8 mm
- Weight: 115 g
- Talk time: up to 3 hours
- Standby time: up to 245 hours
- 2 MegaPixel Camera, (resolution 1600x1200 pixels), video(QCIF)
- Memory expandable using MiniSD card slot
- Twist-on design to access phone, camera, or music functions
- Advanced MP3 Music player
- Bluetooth Compatible
- Tri-band GSM 900, GSM 1800 and GSM 1900
- 10 MB internal memory plus expandable memory up to 1GB for 750 songs with microSD card
- Battery 1100mAh Lition 3 hours talk time, 245 hours standby time

Display and User Interface:

- 176 x 208 pixel TFT active matrix 262,144-color display
- Size: 34.8mm x 41.1mm
- 270-degree rotation (-90 and +180 degrees): Twist on text/phone access, camera, or music functions instantly
- Joystick, two softkeys, application key, edit and clear key, send (Push to talk key) and end key, ITU-T keys, power key
- Music keys: Play/pause, stop, rewind and forward

Nokia 3250 Connectivity:

- Bluetooth wireless technology
    - Talk hands free with a wireless
    - Bluetooth headset enhancement
    - Play wireless multi-player games over a Bluetooth connection
    - Synchronize your phone and compatible PC over a local Bluetooth wireless connection
- USB 2.0 full speed with mass storage profile

Appendix B

The most important characteristics of the Play Station Portable are shown in this appendix.

Figure 26: Play Station Portable (Sony)

- Size 170 mm x 74 mm x 23 mm
- Weight 260 g
- CPU: PSP CPU (clock frequency1,333MHz)
- Memory: 32MB
- DRAM: 4MB
- Capacity: 1.8 GB
- Screen: TFT panoramic (16:9), more than 16 millions of colours and the resolution is 480 x 272 pixels.
- The sound: stereo loudspeakers
- Connections: USB 2.0, wireless 802.11b (Wi-Fi) y Memory Stick PRO Duo, Bluetooth
- Battery**:** Duration depends on functionalities. 10h if you are only listening music, 6h if you are playing and 2-3h if you are watching films.

10. Bibliography

Chapter 1: Introduction

- University of Lancaster Home Page
    - http://www.lancs.ac.uk/
- Infolab21
    - http://www.infolab21.lancs.ac.uk/visitors_to_infolab21/
- Computing Department
    - http://www.comp.lancs.ac.uk/
- Engineering and Physical Sciences Research Council
    - http://www.epsrc.ac.uk/default.htm
- CASIDE
    - http://www.caside.lancs.ac.uk/index.php
- BBC Article
    - http://www.cadenaser.com/articulo/tecnologia/Samsung/arrebata/Motorola/segundo/puesto/ventas/telefonos/moviles/csrcsrpor/20070807csrcsrtec_6/Tes/

Chapter 2: Human Computer Interaction

- Human Computer Interaction (Chapter 2 of the Curricula for HCI of ACM SIGCHI)
    - http://sigchi.org/cdg/cdg2.html#2_1
- Ubiquitous computing
    - http://www-static.cc.gatech.edu/classes/cs6751_97_fall/projects/say-cheese/marcia/mfinal.html
- The context
    - http://www.comp.lancs.ac.uk/~albrecht/pubs/pdf/schmidt_pete_3-2000-implicit-interaction.pdf

- The goals of the HCI
  - http://www.sitepoint.com/article/computer-interaction-site

- Relationship of HCI to other disciplines
  - http://osiris.sunderland.ac.uk/~cs0car/hci/1_des_rel.htm

- Scenarios
  - http://osiris.sunderland.ac.uk/~cs0car/hci/1_des_sce.htm
  - http://www.usabilitynet.org/tools/scenarios.htm

Chapter 3: Usability Engineering

- Centred User design (ISO 13407)

  - http://www.ash-consulting.com/ISO13407.pdf

- The book
  - Nielsen, J. (1993). Usability Engineering. Academic Press Professional, Boston, MA.

- Methods

  - http://www.usabilityfirst.com/methods/index.txl

Chapter 4: Hardware and Software

- Hardware
  - http://www.ti.com/
- Software

  Symbian

  - http://www.symbian.com/developer/

  S60

  - http://www.phonescoop.com/glossary/term.php?gid=129

  - http://forum.nokia.com

  JAVA

  - http://jcp.org/aboutJava/communityprocess/final/jsr082/index.html
  - http://java.com/es/

- Bluetooth
  - http://www.bluetooth.com/bluetooth/
  - http://www.bluetooth.org/spec/
  - http://forum.nokia.com

Chapter 5: Play by Bluetooth

- http://www.forum.nokia.com

Chapter 6: Software Analysis

- http://www.usabilitynet.org/tools/scenarios.htm

- PhD thesis:
  - Toni Granollers. MPIu+a. "*Una Metodología que integra la Ingeniería del Software, la Interacción Persona-Ordenador y la Accesibilidad en el contexto de equipos de desarrollo multidisciplinares*". Universitat de Lleida, July 2004.

Chapter 7: Heuristic Evaluation

- **The book:**
  - J. Lorés. Introducción a la interaccion persona-ordenador. Asociación Interacción Persona-Ordenador, AIPO, 2002.

- **The article:**
  - **"***Appropriating Heuristic Evaluation Methods for Mobile Computing***"** by E. Bertini, T. Catarci, A. Dix, S. Gabrielli, S. Kimani, G. Santucci

Chapter 8: Evaluation

- **Multichannel Adaptive Information System:**
  - http://www.mais-project.it

- **The article:**
  - **"***Appropriating Heuristic Evaluation Methods for Mobile Computing***"** by E. Bertini, T. Catarci, A. Dix, S. Gabrielli, S. Kimani, G. Santucci
- **The Book:**
  - *"Human – Computer Interaction"* by Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale.

Appendix A

http://images.google.co.uk/imgres?imgurl=http://www.mphone.co.uk/nokia/images/nokia_3250_twist.jpg&imgrefurl=http://www.mphone.co.uk/nokia/3250.html&h=241&w=422&sz=18&tbnid=N0akMxIBZVvnLM:&tbnh=72&tbnw=126&prev=/images%3Fq%3Dnokia%2B3250%26um%3D1&start=1&sa=X&oi=images&ct=image&cd=1

Appendix B

http://www.todopsp.com/content/view/98/72/