Universidade do Minho
Escola de Engenharia

José Francisco da Cruz Bernardo

Bluetooth Naming for Situated Interaction in Ubiquitous Environments

Julho de 2009

José Francisco da Cruz Bernardo **Bluetooth Naming for Situated Interaction in Ubiquitous Environments**
UMinho|2009

Universidade do Minho
Escola de Engenharia

José Francisco da Cruz Bernardo

Bluetooth Naming for Situated Interaction in Ubiquitous Environments

A dissertation submitted at University of Minho for the degree of Master in Mobile Systems (Msc)

Under the supervision of
Professor Doutor Rui João Peixoto José

Julho de 2009

Acknowledgements

I would like to thank my advisor, Rui José, for his insight and dedication during the elaboration of this work.

I would also like to thank Pedro Machado, Helder Pinto, Angelo Moreira at Ubisign for the support they provided; Nuno Otero and everyone at Departamento de Sistemas da Informação of Universidade do Minho; Bruno Pinto and Oscar Brito for the help provided, gathering participants for the usability study; participants of the user research studies for their effort; all the people who somehow provided help to me in this work.

Finally, I would like to thank my family, especially Manuela, my mother, and Joana, my companion, for the constant incentive and support.

Abstract

Bluetooth Naming for Situated Interaction in Ubiquitous Environments

This work presents a study over Bluetooth naming and how it may be used as a simple interaction mechanism where people have active control on how they expose themselves and manage their Bluetooth presence. The main goal is to provide a control mechanism built on one's presence and availability.

Bluetooth Extended Naming (BEN), the technique here introduced, enables a self-expression in a way that allows people to use their device name along with additional semantic elements as tags, commands, smileys, etc. In order to support this technique, an investigation effort was taken in order to identify and collect the needs and relevant aspects that would affect its use. Building on this and aiming to fulfill the goal of interacting with public displays, it was specified the grammar of a command language, its language processor developed and integrated on a reactive display control called Instant Places.

The proposed language has been evaluated through several studies of diverse methodology that were conducted along different phases. Initially, a usability study based on questionnaires and procedural interviews was conducted in order to validate the feasibility of BEN as an interaction technique and to determine its limitations and optimizations. A public deployment of Instant Places with limited support of BEN also took place at this phase, for the purpose of collecting logs for analysis. Results of both studies showed no significant limitation, confirming the viability of the adoption of BEN as an interaction technique. In a later phase, a qualitative evaluation study was conducted on the command language usage. Results showed positive feedback from users and indicated a correct and successful usage of the language overall, even though some of the command purpose were not obvious.

Bluetooth Extended Naming demonstrated to be functional and effective as technique for interaction. The results obtained regarding adoption rates and usage observations confirm this. The potential limitations that were anticipated, regarding public rejection and technical difficulties, revealed little impact on its use.

Resumo

Nomeação Bluetooth para Interacção Situada em Ambientes Ubíquos

Este trabalho apresenta um estudo exploratório sobre como a nomeação Bluetooth pode ser utilizada como um mecanismo de interacção simples, que permita às pessoas ter um papel activo no modo pelo qual elas se expõe e gerem a sua presença Bluetooth. O objectivo não é suportar interacção complexa mas criar um mecanismo de controlo construído sobre o estar presente e dísponivel.

A nomeação extendida Bluetooth (BEN), a técnica que este trabalho introduz, pretende disponibilizar um meio de expressão próprio que permita às pessoas utilizarem o nome Bluetooth do seu dispositivo pessoal com elementos adicionais com uma semântica própria, como tags, comandos, smileys, etc. De forma a suportar esta técnica de interacção, foi feito um levantamento sobre aspectos relevantes do uso do Bluetooth e limitações a vários níveis que pudessem condicionar a sua utilização. Sobre este conhecimento foi definida a gramática de uma linguagem de comandos que cobrisse as necessidades e os objectivos da interacção com ecrãs públicos através de BEN. Foi desenvolvido um processador baseado na gramática da ultima versão da linguagem e integrado num sistema reactivo de controlo de ecrãs denominado Instant Places.

A linguagem proposta foi avaliada através de vários estudos de metodologia variada, efectuados em diferentes etapas. Inicialmente foi efectuado um estudo de usabilidade baseado em questionários e entrevistas procedimentais no sentido de validar a viabilidade da nomeação Bluetooth como técnica de interacção e de determinar limitações e possíveis optimizações. Nesta fase, foi também realizada uma instalação pública do sistema Instant Places com suporte limitado de interacção pela nomeação Bluetooth com geração de logs de utilização para análise. Os resultados iniciais demonstraram não haver limitações significativas quer ao nível dos dispositivos quer ao nível da sua utilização para que a técnica proposta seja adoptada. Numa fase posterior, foi efectuado um estudo baseado na avaliação qualitativa do uso da linguagem de comandos definida para suportar o BEN. O feedback obtido foi positivo e na generalidade, os resultados mostraram uma utilização correcta e bem sucedida da linguagem, apesar da finalidade não ser sempre ser óbvia. Observou-se a validação de algumas das decisões de design da linguagem e foram detectados pontos de evolução futura.

A nomeação extendida Bluetooth mostrou ser uma técnica funcional e eficaz para efeitos de interacção. Os resultados relativos à sua adopção e a observações de utilização confirmam-no. As potenciais limitações iniciais revelaram pouco impacto na sua utilização, no que diz respeito a dificuldades técnicas e à rejeição por parte do público.

Contents

List of figures

List of tables

Abbreviations

BEN – Bluetooth Extended Naming
SIG – Special Interest Group
ISM - Industrial, Scientific e Medical
RF – Radio Frequencies
UART - Universal Asynchronous Receiver/Transmitter
USB – Universal Serial Bus
HCI – Host Controller Interface
L2CAP – Logical Link Control and Adaptation Protocol
SDP – Service Discovery Protocol
RFCOMM – Radio Frequency Communication
PPP – Point to Point Protocol
TCP – Transmission Control Protocol
IP – Internet Protocol
OBEX – Object Exchange
PDU – Package Data Unit
IAC – Inquiry Access Code
DIAC – Determined Inquiry Access Code
GIAC – General Inquiry Access Code
IEEE - Institute of Electrical and Electronics Engineers
WI-FI - 802.11 Wireless Networks
URI - Uniform Resource Identifier
RFC – Request for Comments
LL - Leftmost derivation

CHAPTER I

INTRODUCTION

Marc Weiser [1] saw ubiquitous environments as environments embedded with computers and sensors serving multiple individuals; the environment becoming the interface and computation and sensing infrastructures fading into the background. Bluetooth is one technology that can enable such environments.

Bluetooth is a wireless communication technology that has suffered a widespread deployment. Statistics from Bluetooth SIG indicated that up until last year near 2 billion products have been shipped [17]. Bluetooth is becoming a truly ubiquitous technology while it is being integrated in more and more devices of different kinds; headsets, printers, television sets, GPS receivers, and even personal worn items such as watches and sunglasses. However the main reason for this dissemination has been its successful integration with the personal handset.

Bluetooth use is most generally narrowed to the usual set of operations; device pairing, file transfer and synchronization. However, along with these typical functionalities, Bluetooth convenes a set of features that have potential for more than what they were originally planned for. Bluetooth devices have a user-defined name which was created for human readability when exposing to other devices during device discovery. This name can be edited and made public permanently. These operations are normally simple tasks which can be accomplished with the base functionality of any Bluetooth enabled handset and without the need for any specific software.

However, the perception of the general public regarding Bluetooth technology appears to be very biased; it can range from total unawareness to explicit rejection. In the first case, people may even have their Bluetooth radio, on but be totally unaware of the fact that their device is announcing itself to the vicinity. In the second case, people thoughtfully reject the technology based on negative perceptions of the risks associated with surveillance, hacking or spamming. In the middle there is a lot of indifference, by people who are aware of Bluetooth, but when facing the lack of compelling scenarios, do not make a significant use out if it.

The focus of this work is on a technique that uses Bluetooth naming as an interaction mechanism within ubiquitous environments. Although the approach has potential to be used in a multitude of scenarios, in this work it was directed to the use with situated displays. Situated displays are pervasive, public displays that, being equipped with sensing

technologies can capture information about the surrounding environment, and combine information from both physical and virtual worlds in a very effective way. The approach taken extends beyond the simple Bluetooth presence, given by the Bluetooth self-exposure and device scanning mechanisms, and contributes with a technique in which the system can recognize parts of the Bluetooth device name as explicit instructions to trigger actions on the displays. It is called Bluetooth Extended Naming.

Bluetooth Extended Naming main contribution is the definition of a simple command language that regards the undermining limitations of the medium, and while and not aiming to support any complex interactions, structures the possibilities of simple interaction towards situated displays. Despite its technical simplicity one can find this a very promising technique for situated interaction. The use of Bluetooth presence and naming for situated interaction combines very well implicit and explicit forms of interaction. Simply by having a discoverable Bluetooth device, people are already part of the situation and implicitly engaging with the system. By turning the Bluetooth radio on, or changing the device name people engage in explicit interaction with the system.

Making Bluetooth presence visible is also a side contribution of this work. Some people may realize that they unintentionally have their Bluetooth on and may decide to turn it off. Other people may become more aware of what it means and enables, and decide to turn it on and use it. In either case, the end result should be a positive as people will be making more informed decisions and using their personal experiences to decide what is best for them.

1.1 OBJECTIVES

The overall objective of this work is to explore Bluetooth Extended Naming (BEN) as a technique for situated interaction. While this principle can be used in multiple scenarios this work focuses on its use around public displays. For this purpose, we have refined this general objective into the following:

- Define naming guidelines to maximize the efficiency and potential use of BEN.
- Specify a public language for Bluetooth Extended Naming based on requisites, limitations and knowledge acquired from the points above, as well as the language processor to integrate on client systems.
- Determine to what extent BEN can be used effectively as a mechanism for situated interaction by conducting user research studies to inform and support the interaction design process.
- Publish the language to gain feedback and promote usage.

1.2 RESEARCH METHODOLOGY

In this work, the use of Bluetooth naming is explored as a key driver for situated interaction around public displays. The initial concept was obtained through brainstorming over potential and unexploited uses of Bluetooth for interaction purposes. The discussion focused on general interaction requirements of public displays, Bluetooth scanning and presence detection; the concept of Bluetooth naming came forth. In order to explore this concept several research methodologies were devised.

The path initially followed led to the identification of initial limitations and to the materialization of the concept; two simple commands were defined for use in the Bluetooth name and name parsing functionalities were integrated in a prototype that already had a Bluetooth scanning component. This prototype was deployed in a real setting and made public. During its installation, which lasted about seven weeks, the prototype collected data on Bluetooth presence and command usage. After the conclusion of the trial installation, system logs were analyzed. Alongside, a usability study consisting on interviews for performance evaluation on general Bluetooth use and naming practices was conducted. These two studies where helpful in several ways; feedback was obtained, limitations and problems were detected, BEN feasibility was validated and the Bluetooth naming concept evolved through the emergence of new ideas and applications that were derived from this process.

Directions taken afterwards led to definition of naming guidelines which targeted to eliminate, or at least, to minimize the previously detected problems and limitations of the proposed technique. Along with the new ideas that extended the use scenarios and unfolded into new commands, naming guidelines fed the design of BEN command language.

Finally, the outcome of the design process was subject to evaluation, in order to observe command usage and language appropriation. This process consisted on personal, recorded interviews with open-end questions that intended to uncover language design problems, and to extract feedback from users, regarding the full set of commands. The obtained results allowed further improvements to the command language.

1.3 STRUCTURE OF THIS DOCUMENT

This document is organized in the following chapter structure:

- Chapter 2 makes an overview of the Bluetooth technology, its most relevant aspects to this work, as well as reviewing the most relevant pieces of previous research work on the area.

- Chapter 3 introduces Instant Places, the concept and system behind the genesis of BEN.

- Chapter 4 introduces and contextualizes the concept of Bluetooth Extended Naming as a technique for situated interaction.

- Chapter 5 presents user research studies which took place at different phases of this work and, measure and validate BEN as an interaction technique.

- Chapter 6 concludes the dissertation and presents some future work for improvements.

CHAPTER II

STATE OF THE ART

This section gives the reader some background information about Bluetooth and makes the review of related work

2.1 BLUETOOTH TECHNOLOGY OVERVIEW

Bluetooth was given birth in 1994, by an Ericsson Mobile Communications research team headed by Haartsen e Mattisson, working on a short-range and low-cost radio transmission technology for cable replacement in cell phones and accessories. In 1998, several manufacturers gathered to create a Special Interest Group (SIG) for Bluetooth technology promotion and development. By the time, only major technology players like Ericsson, Nokia, IBM, Intel, and Toshiba were represented. Today over 9000 manufacturers have joined Bluetooth SIG, making the technology evolve through critical information and feedback share.

2.1.1 BLUETOOTH SPECIFICATION

A specification is a standard that describes the requirements for implementing a technology and its technical aspects. The main objective of Bluetooth specification is to allow interoperability between compliant devices and applications.

The Bluetooth Special Interest Group is in charge of the development and publication of the Bluetooth specification, a rather complex process where several entities are involved in multiple stages of proposal and revision.

Year	1999	2001	2003	2004	2007	2009
Version	1.0	1.1	1.2	2.0 +EDR	2.1 + EDR	3.0 + HS

Table 1. Bluetooth Specification version releases

The last version of Bluetooth specification (Core v3.0 + HS) [15], is a 1712 page document divided in five volumes, which describe protocol, data transport, component and network topology architectures, as well as the associated terminology and references to the technical language of IEEE.

The following sections describe the Bluetooth specification on its most relevant points of interest to this work. A brief explanation of the Bluetooth architecture is given in order to support the inquiry and device name concepts, the focus of this work.

Architecture

The Bluetooth specification defines a layer protocol stack that implements and allows hardware control. Figure 1 shows the layers that constitute the Bluetooth stack.

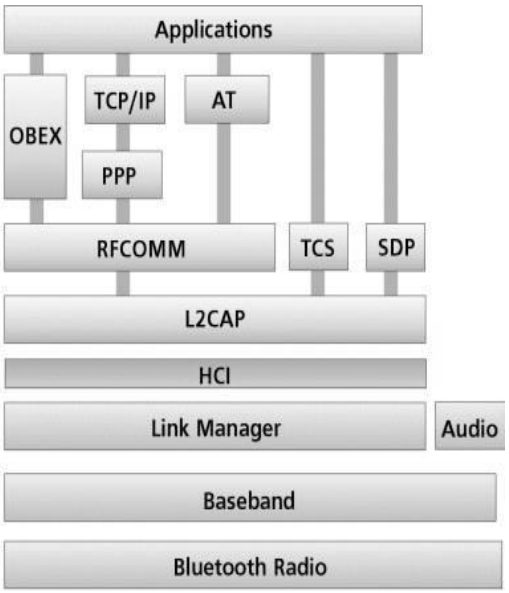


Figure 1. Bluetooth Stack [16]

The Bluetooth stack is divided in two main parts; the host stack which resides as a software component in the host, which can be a pc or a handset; the controller stack which is implemented in the module's chip firmware. These two parts communicate over UART or USB using the Host Controller Interface (HCI) protocol.

The mandatory protocols for all Bluetooth stack implementations are Link Manager Protocol (LMP) and lower layers, L2CAP and SDP. HCI and RFCOMM are protocols which despite being non-mandatory are most often supported, depending on the general architecture. The remaining protocols, PPP, TCP/IP and OBEX are adopted. They are defined by other standards-making organizations and incorporated into Bluetooth's protocol stack.

Bluetooth Radio

The Bluetooth radio layer is the lowest-level of the specification where the requirements for a transceiver (transmitter and receiver) are defined. The Bluetooth radio operates over radio frequencies, in the frequency band at 2.4 GHz ISM (Industrial, Scientific e Medical), which is globally available and intended for unlicensed operation.

A slow hopping frequency scheme is used over the 79 1-MHz slots or channels (23 in certain countries) in which the used frequency spectrum is divided. Hop frequency reaches values around 1600 hops/s and is combined with error correction and verification techniques to insure trustiness of the connection medium.

A Bluetooth radio can be categorized according to power and range. Table 1 shows the correspondence between classes, range and output power.

Class	Range	Power
1	Long (~ 100m)	100 mW
2	Medium (~ 10 - 40m)	2.5 mW
3	Short (~ 10m)	1 mW

Table 2.Bluetooth power classes and ranges [15]

Spectrum Saturation Problems and Coexistence with Wi-Fi

When two wireless technologies share the same medium, coexistence problems may arise. Similarly to Bluetooth, 802.11b technology operates within the 2.4 GHz band, occupying approximately one third of the 2.4 GHz band. Interference can occur when both Bluetooth and 802.11b devices transmit at the same time.

If a Bluetooth network is widespread and under moderate to high levels of utilization, the Bluetooth system will probably offer a substantial number of collisions with an 802.11 network residing in the same area. In this case, Bluetooth and 802.11 would have difficulties coexisting, and performance will likely suffer [21]. The most visible impact is the downgrade of the transmission speed on both technologies. This happens mainly due to the transmission collision of packets, which causes the retransmission of frames and packets, lowering the general data rate.

This lack of coordination between both technologies is well documented [22] and became subject of study of the IEEE 802.15.2 task group. Several practices and techniques have been identified to address this problem.

Baseband

The Bluetooth Baseband is the part of the Bluetooth system that specifies or implements the medium access and physical layer procedures between Bluetooth devices [14]. Basically, this layer defines the mechanisms, entities and procedures that enable the connection and communication between devices.

The communication between Bluetooth devices follows a Master-Slave network topology, called Piconet. Each Bluetooth device is attributed a unique 48-bit device address, obtained from the IEEE Registration Authority. This address is used to determine the frequency hopping pattern that is used by the Bluetooth device when connected in a piconet.

In a piconet, a device takes one of two roles; master or slave. A piconet is characterized for having only one master and up to seven other active participants, synchronized to the same frequency hopping sequence, which is defined over the master device's clock. Additionally, many more slaves can remain connected in a parked state (synchronized, with a physical link but without a logical transport level link established).

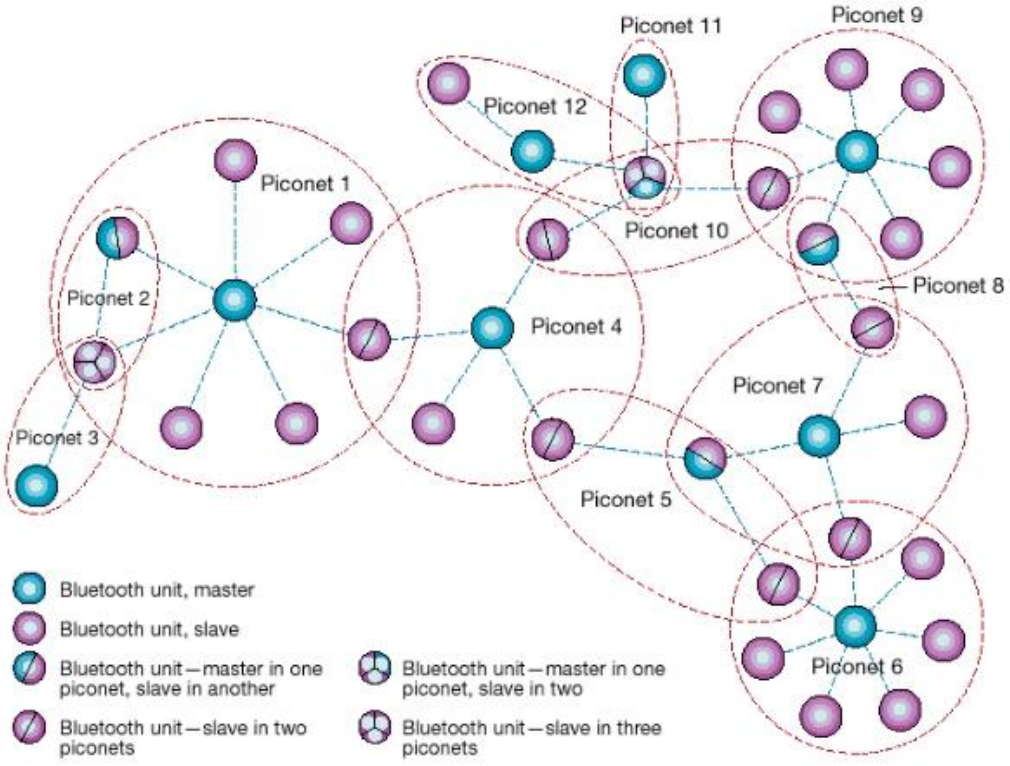


Figure 2. Piconets and Scatternets [16]

The master device shares one physical channel with each device acting as slave, and all communication goes through it, so slave devices do not communicate directly.

There are four Bluetooth physical channels defined, each with a specific purpose. The basic piconet and the adapted piconet channels, used for communication between devices

connected in a specific piconet. The page scan channel, for connection establishment between devices, and finally, the inquiry scan channel, for discovering neighbouring Bluetooth devices. Bluetooth specification assumes that a device is only capable of connecting to one physical channel at any time [14].

Each of the channel specific functions is achieved using simple packets exchange between master and slave devices. The smallest packet size in Bluetooth is the Bluetooth Basic Rate packet data unit (PDU) which is used on the main packet exchange operation. Figure 3 shows the structure of a typical Bluetooth PDU.

Figure 3. The Bluetooth Packet Structure [19]

Each Bluetooth PDU contains three distinct parts; the access code, device identifier, and a payload of data. The access code is used for synchronization and identification. It identifies all packets exchanged on a physical channel: all packets sent in the same physical channel are preceded by the same access code [14]. Depending on the operation that motivated the packet exchange the access code can vary between; device access code (DAC), channel access code (CAC), inquiry access code (IAC). In a typical transaction, packets only include the necessary fields to represent the required layers.

In order control all these resources and determine the behaviour of the device, the baseband has a functional block called Device Manager. It is responsible for every operation of the Bluetooth system that is not directly related to data transport, such as inquiring for the presence of other nearby Bluetooth devices, managing the device local name, making the local Bluetooth device discoverable or connectable, or connecting to other Bluetooth devices, and several other operations implied by higher level layers [18]. In order for these operations to be accomplished the device manager is in coordination with two other core functional blocks; the Baseband Resource Manager, which takes the role of access enabler and scheduler of the physical channels; the Link Controller, which is controlled by the baseband resource manager, and is responsible for carrying out low level operations on the physical channels, such as packet encoding and decoding, flow control, acknowledgement and retransmission

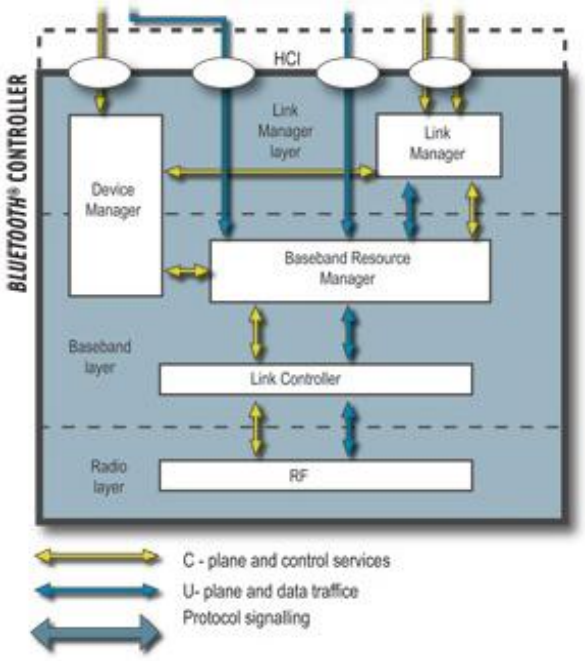


Figure 4. Controller Stack [20]

operations [14].

2.1.2 BLUETOOTH DISCOVERY

Bluetooth enabled devices implement the Generic Access Profile which defines the generic procedures related to device discovery. A Bluetooth device uses the inquiry procedure to discover or to be discovered by other devices in their vicinity. Although master and slave roles are not defined prior to a connection, the term master is used for the inquiring device and slave is used for the inquiry scanning device [14]. The discovery process requires the master to be in the inquiry substate and potential slave devices to be in the inquiry scan substate [Figure 5].

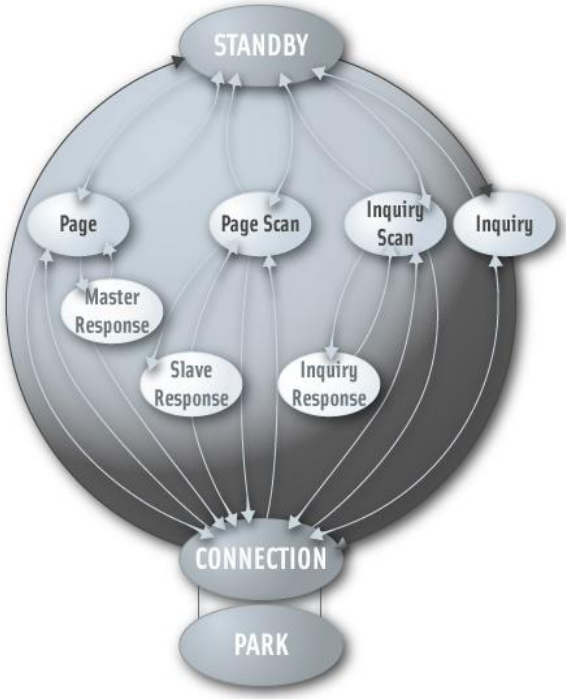


Figure 5. Link controller state diagram [20]

In the inquiry procedure it is used one of the four available physical channels, the inquiry scan channel. The devices involved in this procedure may already be connected to other Bluetooth enabled devices in a piconet, so the inquiry scan channel demands are balanced with the demands of other existing logical transports [14].

The inquiry scan channel follows a slower hopping pattern than the piconet physical channels and which is a short pseudo-random hopping sequence through a 32 channel subset of the RF channels [14]. Through this channel, and in an asymmetrical fashion, the inquiring device which is in the inquiry substate [Figure 4] performs active device discovery. It sends inquiry requests and listens for inquiry responses. Requests are sent iteratively through all possible inquiry scan channel frequencies (hopping) in a pseudo-random fashion, at a very fast rate.

An inquiry request over an inquiry scan physical channel does not require a logical link or any other higher protocol layers. The ID packet type, used in this operation, is very simple and contains the only channel access code (IAC). This code is used to determine how devices will respond to the inquiry; a general IAC (GIAC), will cause all classes of devices to respond; a dedicated IAC (DIAC), which can be one out 63 different codes, will determine the specific class of device to respond.

Discoverable devices, within the broadcast range and in the inquiry scan substate, listen for requests. When a device receives an inquiry message from another device on his inquiry scan channel, it follows the inquiry response procedure and returns a response to the inquiring device; it can return either an Standard Inquiry Response, which uses a packet type

that contains little information like the slave's Bluetooth Address and clock, or an Extended Inquiry Response provides miscellaneous information such as the respondent local name, class of device and supported services.

Name Discovery

During the device discovery procedure, first an inquiry is performed, and then name discovery is done towards some or all of the devices that responded to the inquiry. The name discovery procedure occurs towards slaves that returned Standard Inquiry Response, which does not contain the slave's local name.

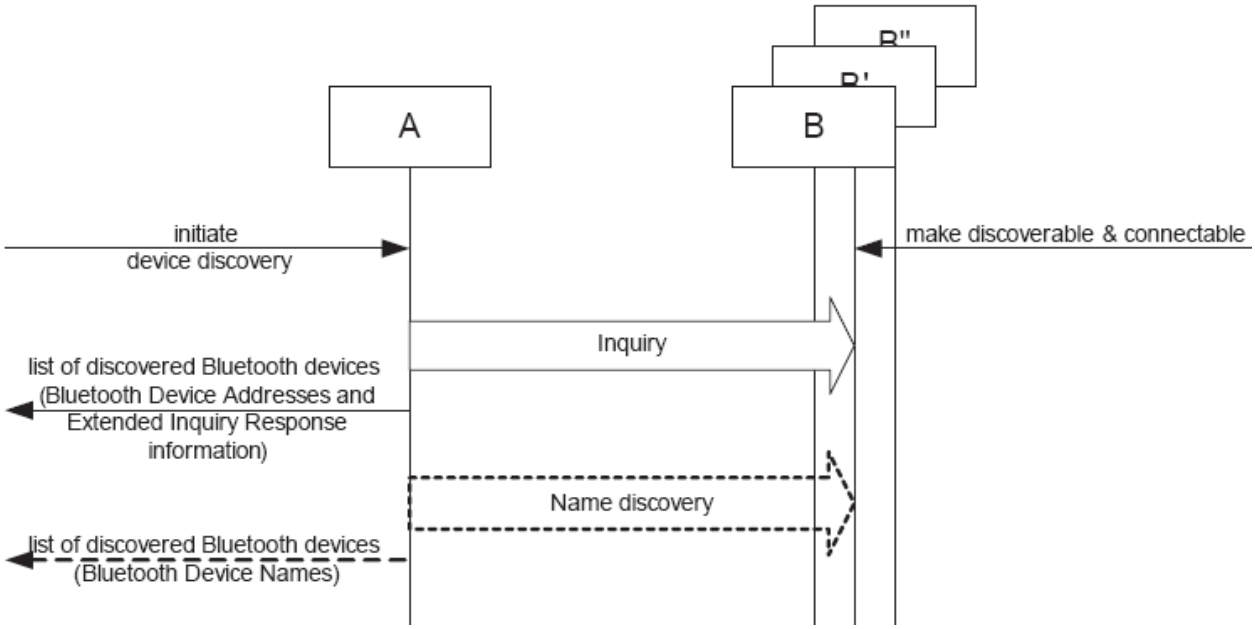


Figure 6. Inquiry and Name discovery [15]

Otherwise, if Extended Inquiry Response succeeds in the inquiry response procedure the name discovery does not occur. This is why sometimes when discovering devices, a representation of the local address shows instead of the device name, at the UI level.

Factors affecting discovery time

According to the specification, the inquiry substate lasts a minimum of 10.24 seconds, unless the inquirer determines to abort the inquiry substate earlier. The inquiry substate uses two 16-channel subsets designated by trains. Each train takes 10 ms to complete and must be repeated 256 times to allow sufficient time to collect all inquiry responses. Moreover, the specification determines that each train must be iterated twice, to ensure that all listening devices in range will be on a common frequency and also in the inquiry scan substate for at least one inquiry time slot; hence the minimum value.

Several factors have been studied and indentified as affecting inquiry time and the overall Bluetooth scanning process.

- Interference from other RF-based technologies such as Wi-Fi, microwave ovens
- Established SCO links (logical transport type) in the inquiry device - a device leaves the inquiry/inquiry scan substates periodically in order to service a synchronous link and is not available to transmit/receive inquiry packets or to send/ listen for responses. Missed inquiry packets cause significant time penalties extending significantly the time of process [14].
- Neighboring devices also in inquiry state - this causes significant delays in the discovery of neighboring devices. It has been shown that the time until inquiry/inquiry scan substates align to allow discovery may be significant [25].

2.1.3 BLUETOOTH STACKS

This section makes a brief presentation of the most significant Bluetooth available for development.

32feet.NET is a shared-source project to make personal area networking technologies such as Bluetooth, Infrared (IrDA), accessible from a .NET programming environment. Bluetooth support requires a device with the Microsoft Bluetooth stack.

BlueZ is an open-source Bluetooth stack which aims to make an implementation of the Bluetooth wireless standards specifications for Linux. It provides support and access for the core Bluetooth layers and protocols. It is flexible, efficient and uses a modular implementation. C is the language to use for application development using BlueZ, thought some Python wrapper available too.

2.2 RELATED WORK

2.2.1 MEROLYN, THE PHONE

"Merolyn, The Phone" [4] is a study of naming practices around Bluetooth which took place in Bristol in August 2006. The focus of this study is to demonstrate the emergent culture around Bluetooth naming of mobile phones and how users appropriated the Bluetooth technology. The approach taken explores these practices on their social, physical and intentional context.

The departure point of this study was the data set of a Bluetooth scanning infrastructure that was deployed in strategic points of Bristol; the city centre, the university campus and a local pub. 1703 names were collected and categorized.

Interview studies were conducted in order to evaluate the contexts in which Bluetooth naming practices took place; the physical context in which the name was set (home, workplace or elsewhere) and the social context (alone, friends, family or work colleagues etc.).

Some of the factors that motivated these practices were also uncovered; the users' awareness of Bluetooth capabilities, the social value given to media sharing between Bluetooth users, the personalization of the name and ambiguity removal when sharing media.

Specific intentions around these practices were sought out in order to understand what was behind the choice of names. Several classes of user intentions were identified: naming the phone, nicknaming the user himself, making a statement, tagging, swearing and identifying the user himself or his group.

This study shows that, in general, the typical uses of Bluetooth like media exchange led to a projection of identity and presence through the Bluetooth channel, parallel to the physical information that flows in human physical engagements or digitally on the internet.

2.2.2 WIRELESS ROPE

The Wireless Rope project [6] is a study of the notion of social context and of the detection of social situations and its effects on group dynamics, based on presence data collected using Bluetooth device discovery. Its main focus is on social contexts where the identities and roles of individuals are disregarded.

The key concept which supports this investigation is that the categorization of the intervenients of typical social engagements may be used for classification of social contexts. This kind of information would be considered as context information that could allow applications to detect meaningful episodes for a user while moving in different social circles and circumstances. In order to accomplish this, intervenients in a social engagement are categorized on their level of familiarity; strangers, familiar strangers and familiar persons. A familiar stranger is a concept fetched from sociology, which is defined by "a person who is

encountered repeatedly, but never interacted" with. System-wise, a familiar stranger is considered to be someone met more than five times.

The Wireless Rope is a mobile application for phones that was developed to collect information of surrounding devices by performing periodic Bluetooth device inquiries. It uses a simple algorithm to distinguish stranger from familiar strangers using previously collected proximity data. A real world deployment was made with a human test subject that carried the application installed in a Bluetooth enabled phone for six days. Results showed differently classified devices and peaks in different social activities in where the test subject was engaged.

2.2.3 REALITY MINING

Reality mining [9] is a study on complex social systems with focus on user's daily activity, information access and use, social patterns and relationships, socially significant locations and organizational rhythms. The key enabler for this study is the data collected from mobile phones which is used to uncover rules and structure in the behaviour of both individuals and organizations.

In order to collect data, a field study was conducted where a system composed of three applications was used. Two of them were mobile applications running on Java enabled phones: BlueAware, which records and timestamps the results of Bluetooth device discovery, and the Context application, which records information such as call logs, cell tower IDs, application usage and phone status (charging and idle). The third application Bluedar ran on a WiFi bridge coupled with a Bluetooth beacon, and also performed cyclic Bluetooth scans sending data over an 802.11b network to the Reality Mining server. The mobile applications were installed in 100 smart phones which were used by participants throughout the time the study lasted.

This study generated data collected by 100 human subjects over 9 months, representing approximately 450,000 h of information about users' location, communication and device usage behaviour.

This fusion of data enabled the extraction of the relationships between subjects correlating proximity, time, and location data. The ultimate goal was to create a predictive classifier that could perceive aspects of a user's life based on this feature.

2.2.4 THE CITYWARE PROJECT

The Cityware project [7] is based on a platform for capturing mobility traces via Bluetooth scanning and has explored several ways of leveraging that information, including a set of in-situ visualizations providing people with information about current or recent Bluetooth presences. The system uses in-situ presence information as a way to generate content for a

Facebook application that lets people associate physical co-presence information with their social network.

2.2.5 PROACTIVE DISPLAYS

The Proactive displays [8] system explores the use of presence as a driver for situated interaction around public displays. The detection of nearby RFID tags triggers the display of profile information about the owner of the Tag, promoting occasional encounters between people around the display. However, this type of approach requires a priori definition of individual profiles with associated data and assumes that everyone will be using a particular type of Tag. Furthermore, people have a very limited role in the system, which is basically to move around and be detected.

2.2.6 MICROFORMATS

According to the official Microformats site [11], the definition of a microformat is "Designed for humans first and machines second, microformats are a set of simple, open data formats built upon existing and widely adopted standards". Microformats are a set of simple, open data formats built upon existing and widely adopted standards like XHTML and HTML to convey metadata.

Several microformats have been developed in order to solve specific problems. Picoformats [12] are an initiative that pursues the spread of this concept to the mobile domain. The goal of the Picoformats project is to use an open process to come up with standards for interacting with mobile services and devices using a SMS-based or command line interface dialects. The use of a picoformat enables the codification of simple commands while satisfying the restrictions imposed by text input on mobile phones. Even though no standards exist to support this, there are some emerging SMS-based dialects supporting interaction with mobile services. Twitter-nanoformats [13] is another initiative that aims to extend twitter capabilities by adding more semantic information to twitter posts through the use of commonly used (or semi-standard) parameters like microformats.

CHAPTER III

CLIENT SYSTEM

Instant Places is an investigation into the role of Bluetooth presence and naming as techniques for situated interaction around public displays [3]. The study specifically addresses the suitability of these techniques and of social practices that emerge in scenarios where they are available. This project motivated the genesis of this dissertation, as support for Bluetooth Extended Naming is integral in Instant Places. This chapter contextualizes the system and how does BEN fit in.

3.1 INSTANT PLACES ARCHITECTURE

The instant places system displays on a public screen content that is directly or indirectly derived from Bluetooth presence information, including content derived from the presence of commands in Bluetooth names [3].

The system is composed by one or more Bluetooth enabled computers each connected to a public screen and linked to a central repository as shown in Figure 5. Information about nearby devices is cyclically collected by a Bluetooth scanner and consumed by a situation data model that manages data about the place and present devices.

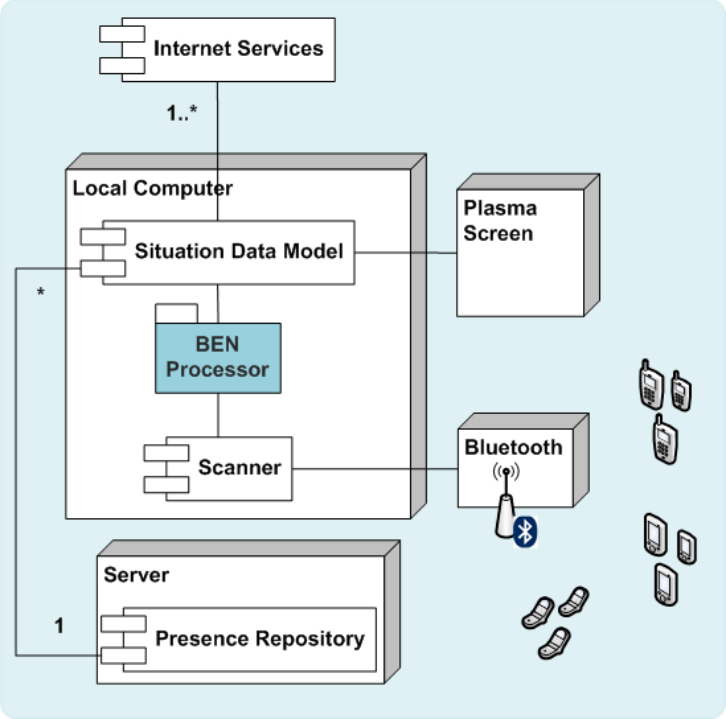


Figure 7. Instant Places Architecture

The central repository maintains persistent information about previous sessions, and combines information from pervasively distributed data sources, allowing for multiple screens in a large space to share the same presence view. The system does not need any a priori information about people, their profiles, permissions or groups, as all the information in the repository is entirely created from the history of presences.

3.2 VISUALIZATIONS

The central functionality supported by Instant Places is the visualization on a public display of content that is situationally relevant and mainly driven by Bluetooth presence information. Figures 6 and 7 show two screenshots of Instant Places that exemplify the dynamic visualizations generated by the system.

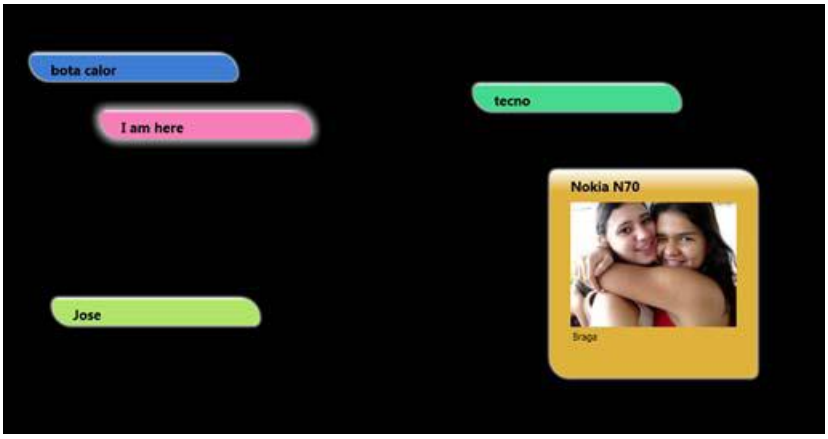


Figure 8. Instant Places Screenshot A [3]

Figure 9. Instant Places Screenshot B [3]

The key driver for the visualization is the real-time information about currently present devices and their names. The periodic scanning of Bluetooth devices generates a continuously changing flow of presence patterns that is visualized on the public display. Each discovered device is represented as a multivariate icon, in which the device name is clearly displayed. Commands included in the device name do not show in the display. This use of Bluetooth names can be classified as an implicit form of interaction in the case of the person who unexpectedly finds his or her name on the display. However, it can quickly turn into an explicit form of interaction when that person changes the device name based on the new meaning created by the visualization on the screen. These elements, per se, guarantee an important base level of content generation and implicit interaction. The dynamic patterns of Bluetooth presence provide an interesting and continuously changing element of situation awareness, and the public display of Bluetooth device names gives them a new meaning and empowers their use. However, they are limited in their ability to produce a continuous flow of enticing content, and that is where extended naming provides the extra level of functionality that invites people to more explicit and engaging forms of interaction [2].

By using Tag or FLK commands in their device names, people are in fact providing seeds that enable the system to retrieve related content from Flickr. A Tag value induces the retrieval and presentation of photos tagged with that word on Flickr service. If a flickr user name is used instead, the system retrieves and displays public photos of the user's Flickr account. These photos are iteratively presented on the neighbouring device associated icon.

This manages to combine situated interaction with a global service to generate a continuously changing display of situationally relevant content [2].

3.3 INTEGRATION AND REQUIREMENTS

Bluetooth Extended Naming is a technique where the Bluetooth device name is used for interaction purposes. The main concept is to use Bluetooth name as command line interface with a command language, a collection of natural-language-derived commands designed with the purpose of enabling simple interaction with near systems, in this particular case with the Instant Places system. Support for Bluetooth Extended Naming is meant to be an integral part of Instant Places. With every scanning operation, Bluetooth devices names are parsed in search for commands and their parameters [3]. The first version of Instant Places implemented a basic, though effective parsing mechanism, that worked with BEN's draft specification. Within the work of this dissertation, it was developed a software component to integrate with Instant Places. The BEN processor is this component which consists in a language processor based on the formal grammar of the command language presented in the following chapter.

In order to accomplish this there are several limitations to consider related to the capabilities of the personal devices, which may undermine its potential, such as the short size of Bluetooth names, limitations in text entry, and also the delay in the detection of name change updates introduced by the timings of the discovery process. A language developed to support expression through this technique should meet several requirements such as simplicity, accessibility and also observe the impact of these limitations.

This section specifies early requirements for the language that supports Bluetooth Extended Naming technique and its integration with Instant Places. These requirements constitute a key issue for this work, as guidelines for the interaction design process, and possibly as set of principles for evaluating future similar works. They will become clearer and more specific further along.

3.3.1 TEXT SIZE LIMITATIONS

Statements in the specification regarding the Bluetooth local name refer to its data type, encoding and length. The specification dictates that the device local name is an UTF-8 encoded string parameter with up to 248 octets in length [1]. With this encoding, each character can have one to four bytes. This means that the number of characters that will make a valid phrase of the language will vary, with the type of characters being used, between 62 and 248 characters.

The limitations imposed by the Bluetooth device itself must also be considered regarding this issue. These limitations are consequence of the device manufacturer implementation. Due to the variety of manufacturers, personal handsets available on the market show different text sizes of device local names, varying greatly from brand to brand.

These two considerations have high impact on language design, inducing the need of strong economy of characters on the language terms, enforcing as much as possible the use of abbreviators or small alias in order to keep the language's expression power.

3.3.2 LIMITATIONS ON THE DEVICE DATA ENTRY MECHANISM

The second criterion introduces the limitations imposed by the Bluetooth device itself at the device data entry mechanism. These limitations are also consequence of the device manufacturer implementation. Due to the variety of manufacturers, personal handsets available on the market offer a rather heterogeneous set of procedures and features regarding data entry at Bluetooth local name. Some of the most used data entry methods follow:

- Upper-Case Alphabet Mode – using this procedure one must interchange between two main modes; the number mode which directly produces numbers; the alphabet mode, which is accessible by pressing the left asterisk (*) key once, and enters a 2 key alphabet mode. Letters are produced by pressing the desired letter on a number key (1) through (9), followed by one of three position keys which selects one of the three key letters: the "left" asterisk (*) key for the left letter, the "middle" zero (0) key for the middle letter or the "right" pound (#) key for the right letter. In this mode, all the letters are in uppercase.

- Shiftable Alphabet Mode – basically the same functionality of the alphabet mode with one significant difference; after the selection of the letter from the triplet, pressing the position selector key again changes the letter case.

- Predictive text – also known as T9 (text on 9 keys), it was developed by Tegic [25]. It combines the groups of letters on each phone key with a fast-access dictionary of

words. It looks up in the dictionary all words corresponding to the sequence of key presses and orders them by frequency of use.

- Multi-Tap/ Multi-press Mode - The most common approach for text input in mobile phones is multi-press. Using this mechanism, the user presses each key a determined number of times to specify the input character. For example, the number 1 key, is pressed once for dot (‘.’), twice for comma (‘,’), the (2) key is pressed once for the character 'A', twice for 'B', and three times for U', and so on. If a desired character is placed at the same key as the previously entered character, like the word “mono”, that has all of its characters on the (6) key, the user must deal with a timeout mechanism, typically between1 and 2 seconds.

The fastest text entry method [26] is the multi-press mode, which might also explain the general predominance of this method in text entry phone interfaces.

Another issue to be considered under this criterion is the character set available for producing the Bluetooth name. Personal handsets available on the market show different character sets, though it is observed that some tendencies have been established by major players and followed by others. Typical alphanumeric characters are usually common among the majority of devices. Special characters, however, like punctuation characters are ma . Most phone use a special character selection grid which is accessible by pressing (*) key, on multi-tap mode, others only allow the introduction of special characters by multi-pressing the (1) key the necessary number of times to produce the character. In either cases and in general, the character accessibility follows a common order based on use frequency, which also translates in proportional physical effort when selecting it for insertion; one key press for dot (.), two key presses for comma (,), three key presses for (?), ...

The impact of these considerations in the language design induces the need of a careful selection of characters regarding their function and frequency of use, and flexibility, allowing that all devices may support it.

3.3.3 SUPPORT FOR KNOWN DATA FORMATS

The third criterion refers to the types and data formats frequently used, and how their syntax should be considered in the language design. Each and every one of these formats has a special syntax, usually defined formally by a RFC. Its inclusion in the language brings an additional challenge while striving to achieve coherence between language terms. The following formats will be considered for use:

- Usernames – websites and web services most often allow users to have an account to be indentified or authenticated by a username and a password. These services have specific requirements on how the username is defined. The union set of all the

constraints determines username expression to be used in the language. Table 3 shows username constraints for some services:

Service	# Characters	Allowed Characters	Forbidden Characters	Further Observations
LastFM	15 max	Alphanumeric	Space + Punctuation	
Twitter	15 max	Alphanumeric	Space + Punctuation	
YouTube	20 max	Alphanumeric		
Facebook	-	Alphanumeric	Punctuation	
Yahoo ID (Flickr,)	4 – 32	Alphanumeric + Underscore + 1 dot		Starting with a letter
Digg	4 – 15	Alphanumeric	Space + Punctuation	
Eventful	2 min			
...	...	...	...	...

Table 3. Service username constraints

- Host or domain names - Hostnames are composed of series of labels concatenated with dots, as are all domain names. Each label must be between 1 and 63 characters long, and the entire hostname has a maximum of 255 characters. RFC 1035 mandate that a hostname's labels may contain only the ASCII letters 'a' through 'z' (case-insensitive), the digits '0' through '9', and the hyphen. Hostname labels cannot begin or end with a hyphen. No other symbols, punctuation characters, or blank spaces are permitted [27].

- Email addresses - An e-mail address, defined in RFC 5322 and RFC 5321, is a string of a ASCII characters subset separated into two parts by an ‘@’ (at sign), a local-part and a domain, i.e., *local-part@domain*. The local-part of an e-mail address can be up to 64 characters and the domain name a maximum of 255 characters [28].
  The local-part may use upper and lowercase characters, digits from 0 to 9, punctuation characters (‘!’, ‘#’, ‘$’, ‘%’, ‘&’, ‘’’, ‘*’, ‘+’, ‘-‘, ‘/’, ‘=’, ‘?’, ‘^’, ‘_’, ‘`’, ‘{‘, ‘|’, ‘}’, ‘~’) and the dot (‘.’), anywhere but on the beginning, end or used two or more times consecutively.
  Although the standards preview the use of this specific set of rules, mail services may impose more or less restrictive ones like allowing case-insensitivity, o restricting to a smaller set of characters, such as alphanumerics along with dot (.), underscore (_) and hyphen (-). Aiming to achieve the minimal common subset for e-mail addresses format, the later set of rules will be adopted here.

- Uniform resource locator (URL) – The URL, defined in RFC 1738, is composed by a combination of the following: the scheme name followed by a colon and then the <scheme-specific-part> whose interpretation depends on the scheme [29]. The scheme name allows digits, and the characters plus ("+"), period ("."), and hyphen ("-") and every alphabetic character.

- Currencies - World currency formats are defined in ISO 4217, the established international standard used in banking and business. Currencies are described by three-letter codes, where the first two letters are ISO 3166-1 alpha-2 country codes and the third is usually the initial of the currency itself. This format eliminates the problem caused by currency names that are common to several countries like dollar, pound or franc [30]. Attending that some devices do not support special characters that represent currencies like '€', '£', '$' or '¥', these provide a less compact but feasible alternative for currency representation.

CHAPTER IV

BLUETOOTH EXTENDED NAMING

Bluetooth Extended Naming is a technique where the Bluetooth device name is used for interaction purposes. The main concept is to use Bluetooth name as command line interface with a command language, a collection of natural-language-derived commands, built with the purpose of enabling simple interaction with near systems. This section defines the language that supports interaction using BEN, and discusses design considerations like syntax and grammar aspects. In order to accomplish a successful interaction through BEN, the user must enter command phrases in the Bluetooth device's local name, which must be syntactically valid within BEN language.

The design process of BEN language was conducted regarding ISO 9241-15 [31], while aiming for a strong reference in ergonomics and coherency. This part of ISO 9241 provides recommendations for the ergonomic design of command languages used in user-computer dialogues. The recommendations cover command language structure and syntax, command representations, input and output considerations, and feedback and help. The domain of application of part 15 of this standard is restricted to command dialogues, and it encompasses command line interfaces, dumb terminals, typically interfaces where user input occurs by recall (they require the use of memory by the user) rather than by selection or direct manipulation.

The kind of interaction in which BEN consists is analogue to some extent to the interaction that takes place between a computer user and a command dialogue. There are fundamental points, however, that make a strong distinction between them, consequences of the different supporting mediums; Bluetooth device name consists of a one-dimensional interface like a command dialog where commands are entered however the output of the actions is observed on a different device; there is no feedback regarding the correctness of the input on Ben, syntactic errors during the command insertion may pass unnoticed; help on BEN is not a direct procedure. Attend these consideration, it is quite obvious that BEN does not fit altogether the requirements ISO 9241-15, however it has a very significant set of features that do fit, and for this, recommendations are considered when applicable.

4.1.1 DEFINITIONS

For the purpose of terminology clarification on this section, the following definitions apply.

Argument

An argument is an independent variable that refers to data on which a command operates.

Command

Text string composed of one or more words, that represents an action requested to the system.

Command Language

Set of commands, structure and syntax that support command-based interaction of user with a system.

Command Stacking

Aggregation of several commands for one-shot input into the system.

Command Syntax

Requirement or rules necessary to build correct phrases with elements of command language.

Command Name

Text string that is used as designator for a command in the command language, and that represents a specific action to be performed by the system.

Command Name Abbreviation

Shortened version of the command name.

Parameter

A parameter is a variable often used with a reserved word to direct the operation of a command.

4.1.2 STRUCTURE AND SYNTAX CONSIDERATIONS

Separators

Most languages, from natural to formal languages, have their phrases, sub-phrases or terms separated by metacharacters. A metacharacter is a special character with a special meaning to a computer program, which overrules its literal meaning, and is typically used for special purposes, either as separator or operator. Metacharacters are the non alphanumeric remainder of the character set. BASH shell reference, for instance, defines a

metacharacter as “a character that, when unquoted, separates words. A metacharacter is a blank or one of the following characters: ‘|’, ‘&’, ‘;’, ‘(’, ‘)’, ‘<’, or ‘>’.” [45].

Whitespace or blank is the per se separator. Command line interfaces, programming languages; the use of whitespace is straightforward. It is used to separate commands and to separate command arguments. However, in this particular context of mobile handsets and their character sets, exceptions exist; the lack of support for whitespace in the Bluetooth local name has been observed in some personal handsets. For what concerns this issue, a need for an alternative syntax imposes, one that may overcome this constraint.

It is proposed here the use of a special syntax, known as dot syntax, for argument separation. The dot syntax is used in some object oriented languages like C#, Java, Action Script in the definition of fully qualified names and member access. For command separation, it is proposed the use of the comma character. The application of the elected separators on the command syntax would be:

[Command Name] **(‘.’)** [Parameter] **(‘.’)** [Argument] **(‘.’)** [Parameter] **(‘.’)** [Argument] **(‘,’)**

Two main advantages support this decision:

- The use of this syntax will succeed in systems where constraints do not allow blanks for separation.
- The dot and comma characters are among the most accessible in most handsets as referred on the previous section. Statistics on phone keyboard data entry methods show clearly that, after alphanumeric and space characters, dot and comma characters have the highest tap rates [26].

Quoting

The introduction of a special character as a separator introduces implicitly the need for quoting. Quoting is a mechanism used to remove the special meaning of some characters and words. It disables the special processing for special characters, and prevents reserved words from being recognized as such. Quoting has a wide usage in programming languages, scripting and command line interfaces. Bash shell, for instance, admits three types of quoting mechanisms: the escape character, single quotes, and double quotes.

The escape character preserves the literal value of the immediately following character.

Single quotes (‘’) preserve the literal value of each character within the quotes. Double quotes preserve the literal value of all characters within the quotes, with exceptions for some characters like the escape character, the quote characters and some others with an associated exception status.

Case Sensitivity

Case sensitivity has been introduced with C language, and has been spread to C-based languages (a significant part of modern programming languages) and command line shells. Case sensitivity reflects in the capitalization of identifiers as a way to encode distinct semantic information. At the present context, speed and comfort at text entry are decisive factors. Changing case would imply changing case mode, which typically requires a few more taps on a control key (often the ('#') key). If compared the overhead taken by this action with the measured gains of case sensitivity, it does not justify its application. Regarding these considerations, and with the absence of other compelling reasons, command names and parameters are implemented as case insensitive in BEN's language.

Opt-out

The term opt-out is traditionally associated with direct marketing campaigns, and refers to methods used to avoid receiving unsolicited product or service information. In this scenario, there are several ways to opt-out, at different levels. One can obviously make his device invisible or turn Bluetooth off, which is device controlled and rather absolute kind of opt-out.

At the language level, the opt-out method is achieved through the no-scan command. This command forces the system to disregard the device at the scanning procedure. One could also introduce invalid commands, but this would be regarded as name and, despite not having direct effect on the system, it would still be considered interaction.

4.1.3 COMMANDS

The following scheme shows commands aggregated in categories according to function. Commands and categories and will be explained in dedicated sections

- Presence Self-Exposure
  - Tag
  - Music
  - Group
  - Id
  - Mood
- Content Submission
  - Name
  - Uri
  - Say
- Actionables

  - Retrieval
    - Get
    - Help
  - Selection
    - Play
  - Assessment
    - Vote
    - Rate
    - Comment

- Opt-Out
  - No Scan
- Proxy
  - LastFM
  - YouTube
  - Facebook
  - LinkedIn
  - Technorati
  - Eventful
  - Twitter
  - Flickr
  - ...

4.1.3.1 PRESENCE SELF-EXPOSURE

Elements of the syntax aggregated under this category are used to express preferences, interests or feelings in a detached way. This kind of command does not regard any particular action or service. Instead, its consequence on the system is non-immediate and diffuse; their assimilation by the system influences its behaviour.

The following table presents self-exposure commands, with associated tokens, syntax and used services. Each one of the following subsections, provide a description for each command and as well as examples of its use.

Command Designation	Service	Tokens	Syntax
Tag	**-**	“Tag”; ”t”	Tag.<*argument*> t.<*argument*>

Music	-	“music”; “m”	music.<argument> m.<argument>
Group	-	“group”; “gr”	group.<*argument*> gr.<*argument*>
Mood	-	Optional “mood”; Emoticon set*	mood.<*emoticon*> <*emoticon*>
Id	Flickr	“id”; “flk”	id.flk.<*argument*>
	LastFM	“id”; “lfm”	Id.lfm.<*argument*>
	YouTube	“id”; “ytb”	Id.ytb.<*argument*>
	Technorati	“id”; “tch”	Id.tch.<*argument*>
	Eventful	“id”; “evt”	Id.evt.<*argument*>

Table 4. Self-exposure commands

Tag, Music and Group commands

In the advent of Web 2.0, associating metadata for characterization of digital information is a consummated trend. This process, generally designated by tagging, is used with many purposes like classification, identification, definition of ownership. The Tag command is here defined for the same purpose; to allow the exposure of user associations regarding any kind of information.

However, the Tag command may be this container, too huge and generic, reason being for the specification of Music and Group commands. These commands specialize the semantics of the Tag command in order to obtain more expressiveness from users. The Music command is to be used for expressing music preferences, like music styles, bands, artists, song names and so on. Likewise, the Group command is meant to express social groups that user belongs to.

Mood Command

The Mood command is used to express emotions or facial expressions. In order to achieve that emoticons are used for values. Emoticons are textual representations of a writer's mood or facial expression, used to complement plain text by disclosing the text temper and

improve its interpretation. Emoticons have been used for quite a long time, and they are integrated and used in instant messengers, web forums, online games, and mobile handsets.

There are two styles of emoticons; Western styled emoticons are read sideways, often from left to right, and have a compact form, using often up to 3 characters. Eastern styled emoticons, used mostly in East Asia, are read normally and are more expensive often using 5 or more characters. The western style of emoticon is, for the sake of character economy, the most convenient for integration with BEN. The following table presents the most frequent used emoticons aggregated by the associated feelings or facial expressions

Happy	Sad	Surprised	Angry	Secretive
:) :-) :-D |-) |-D :->	: (:- < :-c :'-(:-| :-[	:Q :-o 8-O :O 8-|	>:-|):-@):-V >:-<	:-, :-# :-X :-Y :-"
Ironic	**Bored**	**Joking**	**Confused**	**In Love**
:-, :-P :-r	:-\ :-I :-/ |-{	;-) ;-> :*) :-T	:-& :-S :-?	: * E> <3 S2

Table 5. Emoticon set for Mood Command

The presented set of emoticons aggregates a considerable number of alternatives. The intention behind this effort was to close the set for extensions, and to make the emoticon a parameter rather than an argument. This gives some flexibility while allowing an alternative use of the mood command with the command name omitted (only the emoticon), achieving stronger character economy. While assuming this, the use of emoticons as values to other commands must be observed, and in this situation, the quoting mechanism must be enforced to remove their semantic value.

Proposed syntax: **mood . <** *emoticon>*

Synopsis: The mood command uses a text emoticon as parameter, take from a reserved set in order to bring feelings, expressions to the process of self-exposure.

Exs: mood.:)
mood.:(
8|
XD

Id Command

The identification or id command aggregates sub-commands or instances that allow the system to retrieve public information from a specific service associated to a particular user.

The target uses are based on service requests that use the username and do not require authentication. Considering and analysing some representative public services regarding username related features,

- LastFM, via Audioscrobbler service, enables the retrieval of the user's profile info, friend, events, top albums, tracks and much more
- Flickr enables the retrieval a specific internal code, the user's NSID, given the username. The NSID gives access to the user's public information, photos, groups
- YouTube enables the retrieval of the user's public videos and playlists
- The Technorati site allows the retrieval of public information of determined member
- ...

Bluetooth name is public and thus no command could obviously support full authentication and access control without compromising the purpose.

A convenient syntax for this purpose would aggregate in its terms an id command name, the target service label and the username as parameter

Proposed syntax: **id .** *service* **.**<*username*>

Synopsis: Given the *service's* member <*username*>, retrieve the available data associated to this user account and bring it to the process of self exposure

Ex.: id.lfm.frantic0
id.flk.frantic0
id.ytb.frantic0

The examples above enable the retrieval of public information of the member frantic0 for LastFM, Flickr and YouTube services, respectively.

4.1.3.2 CONTENT SUBMISSION

The elements of the content submission type inject or suggest content to the system. Direct content is limited to textual content; other types of content can be submitted by reference. Images, video, web sites, files; all kinds of resources available in Internet are passible of suggestion to the system, using the resource associated Uri.

Command Designation	Tokens	Syntax	Further Observations
Name	No token	<*name*>	
Uri	"uri" "u"	**uri**.<*argument*> **u**.<*argument*>	Quoting
Say	"say" "s"	**say**.<*argument*> **s**.<*argument*>	Sis

Table 6. Content Submission commands

Name Command

The Name command is the minimal command per se. It consists in a simple text string that falls out of the defined command syntax, and that, if positioned isolated or before any command in the Bluetooth name, will be considered the user's self-intituled name.

Uri Command

The Uri command is used to submit content to the system by reference. The argument type of this command is an Uri that may reference and kind of resources available online, like images, video, sound files, web sites and that the user wishes to introduce in the system.

The use of this data format introduces the need to cope with its intrinsic separators. These separators are special characters within BEN's language must be overridden with the use of quotes.

Say Command

The Say command is a text submission type that is used for nano-blogging. After the huge success of Twitter [32], a micro blogging online service where messages (tweets) are restricted to a maximum of 160 characters in size, new on-line services pursuing the same trend, like Adocu [33] which introduces the nano-blogging concept where messages are restricted to one single word.

The maximum length of the Say command argument is obviously limited to the available characters left in the Bluetooth name. In order to maintain coherence with the general

language, a say message that exceeds a single word must have a special character separating words if device constraints determine that blanks are usable. Consequently, the argument must be contained between quotes, both for argument delimitation and, if applied, to ignore the special value of separators. If Say is to be used in a whitespace non-supporting phone, words in Say messages must separated by dots.

4.1.3.3 ACTIONABLES

The commands of this category are used in a stimulus-response context. The system stimulates user's responses by displaying engaging content or information.

Command Designation	Tokens	Operators	Syntax
Get	"get" "g"	-	**get .**<*reference*> **g .**<*reference*>
Help	"help" "h"	-	**help.** **h.**
Play	"play" "p"	-	**play.**<*reference*> **p.**<*reference*>
Vote	"vote" "vt"	none: vote on (**+**): vote for (**-**): vote against	**vote.**<*reference*> **vt .+.**<*reference*> **vt . - .**<*reference*>
Rate	"rate" "rt"	System-defined numeric range [1-5], [1-10], [0-100]	**rate.**<value>**.**<*reference*> **rt .**<value>**.**<*reference*>
Comment	"comment" "c"	-	**comment .**<*reference*> **c.**<*reference*>

Table 7. Actionable commands

RETRIEVAL COMMANDS

Get

The get command is used for the retrieval of local available media. The media items which are shown by the system may induce the user to download them to their personal handset. In order to achieve that, the media items exposed by the system must be documented with a unique reference that is used as argument of the command. The get command's action triggers an OBEX push in response containing the media item.

Proposed syntax:	***get .*** *< transaction_reference >* *or* ***g .*** *< transaction_reference >*
Synopsis:	Triggers an OBEX push reply containing the media item referenced by *transaction reference*
Ex.:	get.item45 get.ringtone23 get.t43 get.g34

Help

The *help* command provides help by enabling the retrieval of a small text file with the list of available commands and their syntax .

Proposed syntax:	**help.** or **h.**
Synopsis:	Triggers an OBEX push reply containing the a text file with available commands and respective syntax
Ex.:	help.

SELECTION COMMANDS

Play

The Play command is used to request the presentation of available media on the display. Presenting the media means video or audio playback, including a picture in a slide show, or showing it on full screen mode. Like in the previous command, a unique reference associated to the media item is used as argument of the command.

Proposed syntax:	***play .*** *[<item reference>]*

or

p . *[<item reference>]*

Synopsis: Triggers the presentation of the media referenced by *item reference*

Ex.: play.#12
p.i007

ASSESSMENT COMMANDS

Assessment commands fall in the actionables category, as they are to be used in situations in which the user finds certain content items, displayed by the system, engaging and appealing for him to emit statements of opinion or support. In order to direct the command action to the item of interest, a mechanism of item referral is applied. Each item displayed by the system is documented by a unique reference. The user must use the reference in the command along with a specific operators and arguments.

Vote

The *vote* command enables the expression of users' decision or opinion regarding a specific subject. The By allowing an optional operator, Vote command supports two voting systems; the single-winner system, which fundamentally consists in the selection of one from many, and the approval system, which determines something to be approved or not. The inclusion of the operator supports the latter voting system.

Proposed syntax: **vote .** *operator?* **.** *[<vote subject> | <item reference>]*

or

vt . *operator?* **.** *[<vote subject> | <item reference>]*

Synopsis: Votes on/for/against the subject referenced by <item reference>

Ex.: vote **.** #s32
Vote **.** + **.** Peace
vt **.** - **.** 123

Rate

The r*ate* command enables the public evaluation regarding a specific subject in terms of a defined range.

Proposed Syntax: **rate**.*<range>.[<rate subject>| <item reference>]*
or
rt.*<range>.[<rate subject>| <item reference>]*

Synopsis: Rates the subject referenced by <*item reference*> by a value contained in the range.

Exs: rate.1."http://www.twitter.com/"
rt.5.the dark knight
rt.9.i003

Comment

The *comment* command is used for commenting a specific subject displayed by the system as an item with a unique reference. The argument takes free text as argument and its length is determined by the available characters left in the Bluetooth name. It must also be contained between quotes.

Proposed Syntax: **comment . [** *<comment subject>| <item reference>* **] . <***message* **>**
or
c . *[<comment subject>| <item reference>]* **. <***message* **>**

Synopsis: Comments the subject by label or by <item reference>

Exs: c **.** #123."Loved it. will have it again"
c **.** 001."Expecting more"

PROXY COMMANDS

Proxy syntax elements are inspired in well-known online services with public APIs that are relevant to integrate with the system. While aiming to achieve this, one must attend to the

necessity of creating a flexible mechanism that allows the integration new and future services. Considering this, Proxy commands category is meant to be an extensibility mechanism that allows for the integration of new services. Commands of this category are differentiated by follows a different syntax. The keyword which indicates the command name is omitted here. All proxy commands start directly with a dot.

Commands considered relevant to incorporate in the language must be defined regarding the features that service APIs provide, like interaction requirements and necessary data request formats.

Command Designation	Tokens	Syntax	Example
LastFM	"lastfm"; "lfm"	**.lastfm .** [*arguments]* **.lfm .** [arguments]	**.lfm.**metallica
Flickr	"flickr"; "flk"	**.flickr .** [*arguments*] **.flk .** [*arguments*]	**.flk.**beach
YouTube	"youtube"; "ytb"	**.youtube.** [*arguments*] **.flk .** [*arguments*]	**.ytb.**"obama fly "
Eventful	**"**eventful**";** **"**evt**"**	**.eventful.** [*arguments*] **.evt.** [*arguments*]	**.evt.**braga
Twitter	**"**twitter**";** **"**tw**";**	**.twitter.** [*arguments*] **.tw .** [*arguments*]	**.twt.**frantic0
…	…	…	…

Table 8. Proxy commands

4.2 PROPOSED GRAMMAR

This section presents the grammar that describes the specified command language. Several of aspects defined and specified before are here considered for structuring of the grammar. Lexical rules are defined over the command names, their arguments and accepted values like usernames, email, urls, operators, and free text. Syntactic rules are defined based on the specified overall structure of the language.

4.2.1 TERMINALS AND REGULAR EXPRESSIONS

The terminal symbols of the grammar have been defined on account of values defined in the naming guidelines section relatively to known data formats. Terminals here considered are the command names and abbreviations and values for the command arguments, such as separators, language operators and known data formats that can be used. For the latest, which consist in more structured data formats, like emoticons, URIs, email addresses, quoted strings and free text, complex regular expressions have been used. Regarding command names and services, terminal symbols were defined to enable case insensitive names and abbreviations, and one terminal was defined for each of the command name forms.

4.2.2 NON-TERMINALS AND PRODUCTION RULES

The defined grammar allows the use of an optional identifier, for the user's nickname, followed by a list of commands. Each command may begin with a command name from the proposed set or, with a dot if it is a proxy command. The production rules where defined allow whitespace skipping, to prevent the syntactic errors that happened on the trial study. Regarding the whitespace incompatible devices, the grammar was defined to allow commas to be used in name-command separation, and in command-command separation. This way, a user with a device without whitespace support can also use a name and a list of commands. Emoticons were implemented directly as a base command in the grammar to enable their use as nameless mood command. The mood command that follows the general syntax is also available. Quoting has been implemented to support multi-word values like email addressed, URIs and free text and enable the disregard of special characters.

```
// Non-terminals & production rules

Start   -> Name? COMMA? Command* EOF;

Command -> Keyword? ( DOT Postfix )* COMMA?
        | EMOTICON
        ;

Postfix ->   Service
        |    Operator
        |    Range
        |    Value
        ;

Operator -> PLUS |  MINUS
         ;

Name    -> ALPHANUM_IDENTIFIER;

Value   -> DOUBLE_QUOTE Quoted_Value DOUBLE_QUOTE
        | UnQuoted_Value
        ;

Quoted_Value -> EMAIL_ADDRESS
              | URI
              | EMOTICON
              | FREE_TXT
              ;

UnQuoted_Value -> GEN_IDENTIFIER
              | EMOTICON
              ;

Service ->     Flickr
        |      LastFm
        |      YouTube
        |      Twitter
        |      Eventful
        |      Technorati
        ;

  Function ->  ID
        |      Tag
        |      Music
        |      Group
        |      Get
        |      MOOD
        |      Rate
        |      Vote
        |      Comment
        |      Play
        |      Say
        |      Help
        ;
```

```
Tag     -> TAG_A | TAG_F;
Music   -> MUSIC_A | MUSIC_F;
Group   -> GROUP_A | GROUP_F;
Rate    -> RATE_A   | RATE_F;
Comment -> COMMENT_A | COMMENT_F;
Get     -> GET_A | GET_F;
Vote    -> VOTE_A | VOTE_F;
Help    -> HELP_A | HELP_F;
Say     -> SAY_A | SAY_F;
Play    -> PLAY_A | PLAY_F;
Flickr  -> FLICKR_A  | FLICKR_F;
YouTube -> YOUTUBE_A | YOUTUBE_F;
LastFm  -> LASTFM_A | LASTFM_F ;
Twitter -> TWITTER_A | TWITTER_F;
Eventful -> EVENTFUL_A | EVENTFUL_F;
Technorati -> TECHNORATI_A | TECHNORATI_F;


/* Terminals & Regular expressions */

EOF                  -> @"^\s*$";
[Skip] WHITESPACE    -> @"\s+";
DOUBLE_QUOTE         -> @"\""";
PLUS                 -> @"\+";
MINUS                -> @"-";
DOT                  -> @"\.";
COMMA                -> @",";
RANGE2               -> @"[0-9]{2}";
RANGE1               -> @"[0-9]{1}";
EMOTICON             -> @"[O>}\])]?[:;8D=X][',]?-?[:A-Sa-
s#@*$|3<>\\\\\)\]}]";
URI                  -> @"^(http\:\/\/[a-zA-Z0-9_\-]+(?:\.[a-zA-Z0-9_\-
]+)*\.[a-zA-Z]{2,4}(?:\/[a-zA-Z0-9_]+)*(?:\/[a-zA-Z0-9_]+\.[a-zA-Z]{2,4}(?:\?[a-
zA-Z0-9_]+\=[a-zA-Z0-9_]+)?)?(?:\&[a-zA-Z0-9_]+\=[a-zA-Z0-9_]+)*)";
EMAIL_ADDRESS        -> @"^[\w\.\-]+@[a-zA-Z0-9\-]+(\.[a-zA-Z0-9\-
]{1,})*(\.[a-zA-Z]{2,3}){1,2}";
QUOTED_STRING  -> @"@?\""(\""\""|[^\""])*\""";
ALPHANUM_IDENTIFIER  -> @"[a-zA-Z0-9_]*";
GEN_IDENTIFIER -> @"([0-9A-Za-z_:;-]|\(|\))*";
FREE_TXT             -> @"([\sa-zA-Z0-9_:;,-]|\?|\!|\.|\(|\))*";
```

```
// Command & Service KEYWORDS

ID       -> @"[iI][dD]";
TAG_F    -> @"[tT][aA][gG]";
TAG_A    -> @"[tT]{1}";
MUSIC_F  -> @"[mM][uU][sS][iI][cC]";
MUSIC_A  -> @"[mM]{1}";
GROUP_F  -> @"[gG][rR][oO][uU][pP]";
GROUP_A  -> @"[gG][rR]";
MOOD     -> @"[mM][oO][oO][dD]";
GET_F    -> @"[gG][eE][tT]";
GET_A    -> @"[gG]{1}";
PLAY_F   -> @"[pP][lL][aA][yY]";
PLAY_A   -> @"[pP]{1}";
SAY_F -> @"[sS][aA][yY]";
SAY_A -> @"[sS]{1}";
HELP_F -> @"[hH][eE][lL][pP]";
HELP_A -> @"[hH]{1}";
VOTE_F -> @"[vV][oO][tT][eE]";
LASTFM_F -> @"[lL][aA][sS][tT][fF][mM]";
LASTFM_A -> @"[lL][fF][mM]";
YOUTUBE_F -> @"[yY][oO][uU][tT][uU][bB][eE]";
YOUTUBE_A -> @"[yY][tT][bB]";
TWITTER_F -> @"[tT][wW][iI][tT][tT][eE][rR]";
TWITTER_A -> @"[tT][wW][tT]";
EVENTFUL_F -> @"[eE][vV][eE][nN][tT][fF][uU][lL]";
EVENTFUL_A -> @"[eE][vV][tT]";
TECHNORATI_F -> @"[tT][eE][cC][hH][nN][oO][rR][aA][tT][iI]";
TECHNORATI_A -> @"[tT][cC][hH]";
VOTE_A   -> @"[vV][tT]";
RATE_F -> @"[rR][aA][tT][eE]";
RATE_A   -> @"[rR][tT]";
COMMENT_F -> @"[cC][oO][mM][mM][eE][nN][tT]";
COMMENT_A -> @"[cC]{1}";
FLICKR_F -> @"[fF][lL][iI][cC][kK][rR]";
FLICKR_A -> @"[fF][lL][kK]";
EVENTFUL_A -> @"[eE][vV][tT]";
EVENTFUL_F -> @"[eE][vV][eE][nN][tT][fF][uU][lL]";
```

BEN Processor Architecture

The grammar of the command language was specified as a LL(1) grammar in order to fed to TinyPg [36], a parser generator. This powerful utility produces LL(1) recursive descent parsers given a LL(1) grammar for input, by generating source code directly as output. The output of TinyPg, which is generated from the specified language, consists on a partial implementation of BEN processor. It contains three classes [Figure 10]; the scanner, the parser and the parse tree. In the proposed architecture, the scanner is an internal

component of the parser which performs lexical analysis on given inputs, that is, BEN expressions obtained from Bluetooth scanning. This process consists on the conversion of the input, a sequence of characters, into a sequence of tokens.

The parser is the component that performs syntactic analysis of the sequence of tokens produced by the scanner, to determine its grammatical structure and validate it against BEN formal grammar. The Parse method of this component encompasses both the lexical and syntactic analysis procedures and stores the final resulting parse tree in the Parse Tree component.

The Parse Tree class has the Eval method where semantic evaluation has been implemented. Semantic evaluation causes the traversal of parse tree, and stores its data in a BEName object, a friendlier data structure to be consumed by the client system's data model.

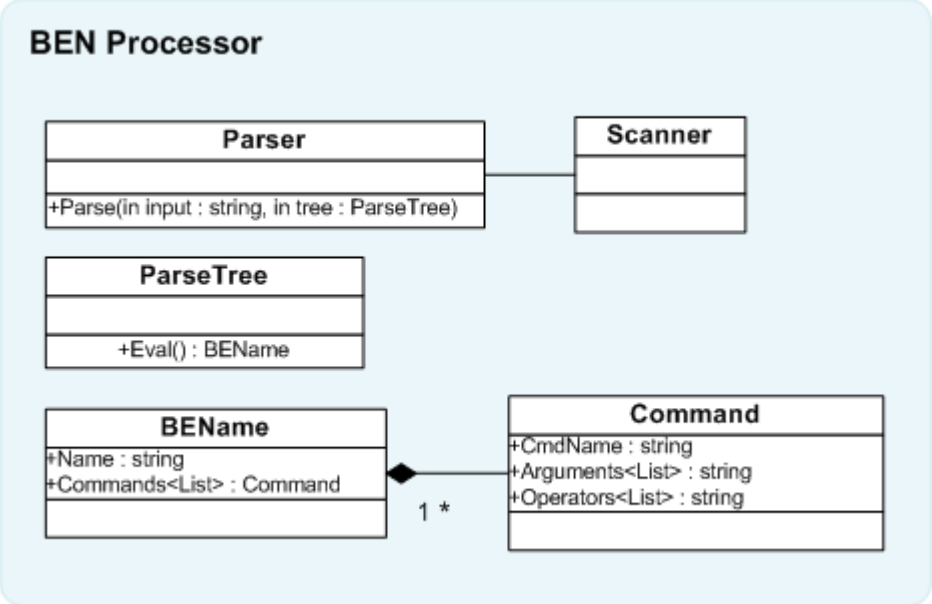


Figure 10. BEN Processor Architecture

CHAPTER V

USER EVALUATION

This chapter presents a set of user research studies that were performed at different stages of this work. At an initial phase, it was taken a study that combined a set usability interviews with a trial installation of Instant Places in a public bar. This study followed a quantitative approach in order measure practices around Bluetooth in general and device naming in particular.

The final study presented in this chapter aimed to evaluate and produce a description of the dynamics of the interaction process with the final version of BEN's command language. The interaction process involves a more complex human behavior, more difficult to be captured with quantitative techniques and thus a qualitative approach has been taken.

5.1 USABILITY STUDY

The purpose of this initial study was to obtain a preliminary insight of procedural mechanisms associated with the use of Bluetooth naming. The chosen methodology was a series of usability interviews with mobile phone owners to evaluate the performance in the execution of tasks related with Bluetooth naming and uncover any limitations or suggestions related with the use of Bluetooth naming.

Interviews were conducted with 40 participants (26 male and 14 female), aged between 15 and 28 who said to have had some type of Bluetooth usage before. The interview was divided in two distinct parts: a survey on general Bluetooth use and naming practices that was answered by all the participants; and a set of 5 tasks related with Bluetooth naming that were executed by 12 of the participants. The tasks were executed with the participants own phones, and the objective was to measure performance and highlight any factors that could affect their execution.

5.1.1 SURVEY RESULTS

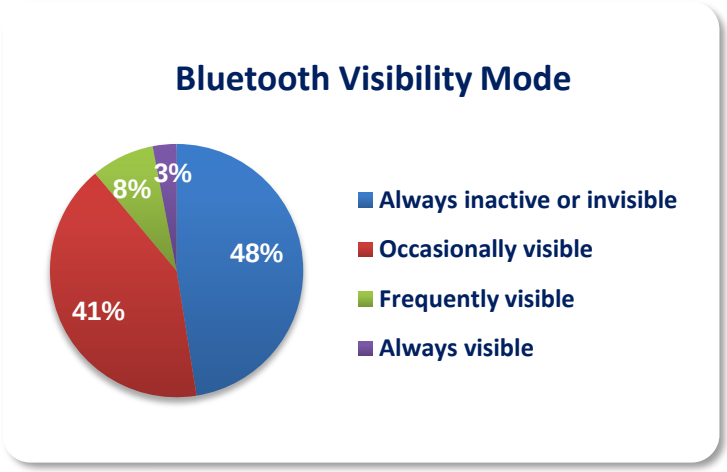


Figure 11. Bluetooth Visibility Mode

The first group of questions was about the visibility mode used in the Bluetooth device. 47% of the responds indicated always having their Bluetooth inactive or invisible. 41% responded that they occasionally had their device visible, 8% that they frequently had their device visible and 3% that they always had their device visible. The reasons for not being visible were most of all related with fears of attacks on the personal device, such as the possibility of infection by virus. Many people also mentioned a possible loss of privacy and saving energy (many respondents seemed to be aware of the consequence of Bluetooth on battery life).

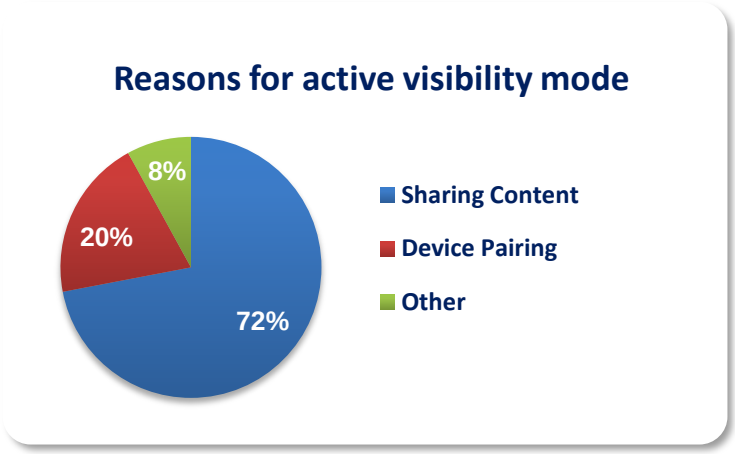


Figure 12. Reasons for active visibility

The reasons for activating Bluetooth and making it visible, were essentially sharing content (72%) and device pairing (20%). Some people specifically mentioned making use of "temporary visibility", a feature available in some devices that allows visibility to be activated only for a limited period of time.

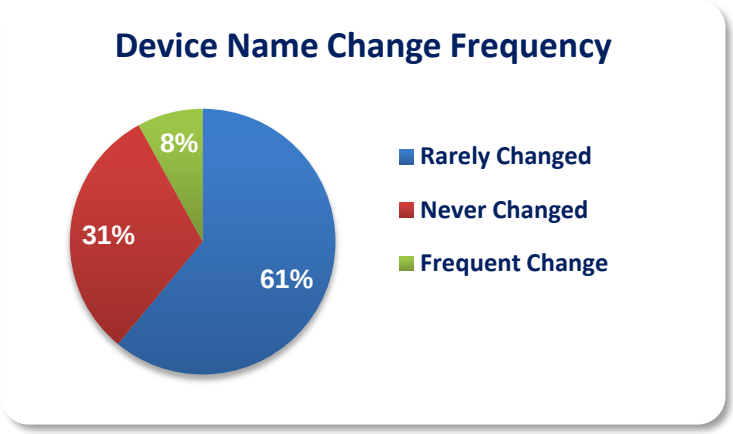


Figure 13. Name Change Frequency

Regarding Bluetooth naming practices, participants were asked how often they changed their Bluetooth device name. The vast majority answered that they rarely changed their name (61%) or even that they had never changed their device name (31%). Only 8% reported changing their device name occasionally or frequently.

On average, participants indicated being able to write device names with 31 characters on their mobile phone, with 16 being the minimum and 65 the maximum.

5.1.2 TASK ANALYSIS

In the second part of the interview, participants were asked to perform the following 5 tasks:

1. Change the current activation state of their Bluetooth device.
2. Change their device name (new name had to be realistic)
3. Introduce a command in the device name (CMD:activate)
4. Introduce a command with two parameters (CMD:activate,num)
5. Introduce 2 commands, each with a parameter (CMD:activate CMD:num)

5.1.2.1 RESULTS

Figure 14 presents graphics that summarize the performance results.

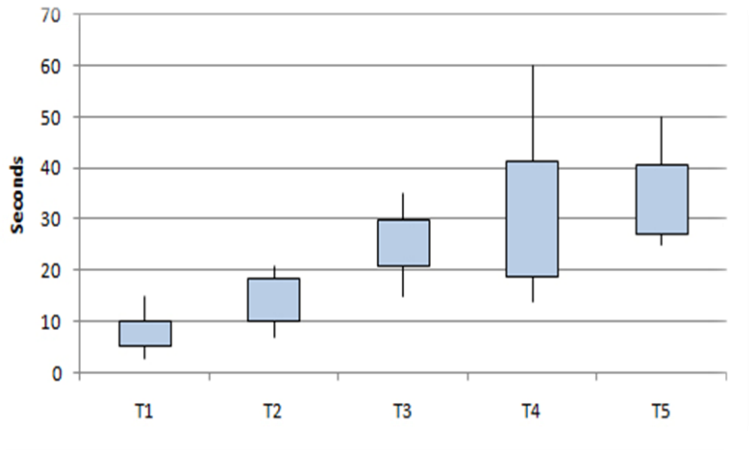


Figure 14. Min, 1st Quartile, 3rd Quartile and Max times for task execution

Results for tasks 1 and 2 indicate a generally good performance in the most basic operations of activating/deactivating Bluetooth (T1) and changing device name (T2). Particularly good results were obtained by participants with speed dial functionality already associated with Bluetooth activation/deactivation. These are tasks that can be easily and quickly achieved, even while on the move.

The introduction of commands in device names (T3, T4 and T5) takes too long to be something that the average user would do without some disruption to other tasks. However, the average duration (about 30 seconds) is perfectly suitable for someone not immediately involved with other tasks, and these were tasks that people were doing for the first time. One of the participants was not able to complete tasks 3, 4 and 5 because his device did not support whitespaces as part of the name.

Participants also made some suggestions on how to improve the performance of these tasks, such as using only lower case letters and replacing the use of the ":" character by the dot character ".", more easily accessible on most mobile phones.

5.2 REAL WORLD DEPLOYMENT

This section presents the results obtained from data collected during the deployment of the Instant Places prototype in a bar at the University of Minho campus [José et al, 08]. This place was elected considering the amount of people that usually go there for coffee or a quick snack, normally in small groups.

The study involved two sequential phases:

- Silent scanning – initially a silent Bluetooth scanning was conducted to obtain a neutral perspective of the local Bluetooth environment. This phase lasted four weeks;

- Public Deployment – the system went public with a specific visualization, and leaflets with information about the project and instructions about the use of commands in Bluetooth names were distributed. This phase lasted three weeks;

The prototype integrated support for the draft version of BEN specification which defined two types of commands: Tag and Flickr. Command names and parameters were separated by a colon (':'); parameters were separated by commas (','). The Tag command was used to associate multiple tags, and was activated by including in the name the expression "*Tag:*" followed by a comma separated list of tags, as in the following example "*my device Tag:punk,rock*". The Flickr command associated a Flickr user name with the device and was activated by including in the device name the expression "*flk:*" followed by the flickr user name, as in the following example "*mydevice flk:JohnSmith*". In both cases, these commands were intended to serve as hints for the display of photo streams obtained from the photo sharing website Flickr.

5.2.1 LOG ANALYSIS

The logs collected by Instant Places enabled the extraction of data regarding the device addresses and unique devices names were seen during these two periods. The total number of visits to the bar was based on sales numbers. Table 1 compares key Bluetooth usage parameters between the two phases.

	Silent Scanning	Instant Places
Estimated visits	7625	6526
Unique devices	365	460
Unique names	317	685
% visits with BT visible	4.7%	7.0%
Names per device	0.9	1.5
Names with commands	n.a	112

Table 9. Effect on Bluetooth usage patterns

The overall results of the trial indicate that the proposed techniques were easily and widely adopted as part of situated interactions around a public display.

Regarding BEN usage, there were 112 occurrences of names that included Tag commands. From these, some uncovered difficulties in command usage and were analyzed. The results are summarized in Table 5.

	Type	Occurrences
Names with space after Tag	Syntax error	8
Names with space between multiple tags	Syntax error	5
Names without space before Tag	Syntax error	3
Names only with tags	Ambiguous	6
Names with Tag: but no Tag specified	Ambiguous	2
Names with Tag: repeated	Ambiguous	1

Table 10. Issues on the Tag command usage

There were 16 cases with at least one syntax error that prevented the successful indication of tags. This consisted in 14.2% of those who tried use the Tag command in their name. In 5 of those errors, the same name was observed with the correct syntax, indicating that after a first failed attempt, 5 people managed correct their syntax error and produce valid commands. This consideration lowers the failure rate to 9.8%.

There were also 9 ambiguous names that were not necessarily incorrect but were not anticipated in that version of BEN specification. In 6 of them, the Tag command was the name. This was not properly parsed because the first word of the name was used as an identifier that was not considered for command parsing.

There were 2 names without any Tag specified, even though the “Tag:” expression was present. Interestingly, this may have not been an error, but an emerging strategy for facilitating at a later point the recollection of the Tag syntax, or reducing the input needed for adding new tags. One name had the “Tag:” expression repeated, which in the current parser would ignore the second expression.

These included 5 tags explicitly naming University degrees, 6 related to football clubs, and 3 trying to re-direct attention to other blogs. The third relevant category (14%) was tags that could be considered obscene or satirical. In most cases these were clear attempts of “winning” over the limitations imposed by the system by causing the display of inappropriate photos. In the interviews, respondents considered that their use of tags was not really part of a thoughtful strategy to combine device names and tags.

5.3 QUALITATIVE RESEARCH

In this section, it is presented a qualitative evaluation of the use of the command language that supports BEN. The focus of this evaluation is the outcome of an intermediate step of the language design process described in chapter 3, which was based on knowledge acquired from the two preceding studies, naming guidelines and further reasoning regarding specific interaction needs. The version of the command language under this evaluation was composed of the following commands; Tag, Music, Group, Id, Mood, Name, Uri, Shout, Get, Help, Show, Vote, Rate and Comment.

The goals for this evaluation were:

- Introduce the interaction process and validate users' conceptual model
- Observe the command set usage
- Extract general feedback from users

Ideally, the kind study to be conducted at this stage would have been a complete field study where a prototype of Instant Places with full BEN support, and associated visual feedback, would be placed in realistic settings for usage observation. However, given time schedule and resource constraints, this was not viable. It was thus decided to learn the most from the user's perspective of the command set and its use within the interaction process on several proposed scenarios. This allowed the exploration of the full set of commands which would hardly occur in real settings.

5.3.1 TARGET AUDIENCE

In order to obtain relevant results with this process, the choice of an appropriate target audience is vital. To determine the target audience it was considered that it would be composed of people who would like to interact with public displays using BEN. The following factors were considered in the profile to be matched by participants:

- Age – people located in the age interval [20 - 55]
- Technology acquaintance – people that have a Bluetooth device and are well acquainted with its configuration and features use.
- Use profiles – people that have used the mobile device's Bluetooth features several times and are comfortable typing text using the mobile phones keyboard.

Nielsen [34] suggests that the best results come from testing no less than 5 users, and running an affordable number of tests with them. According to him, this would grant a minimum level of 85% of effectiveness to the study, and that the significance of each additional user would contribute less to increase this value. While facing time and resource constraints, the interview was conducted with 12 people (7 males and 5 females).

5.3.2 METHOD

The user study was composed of individual evaluation sessions, with audio recording, carried out in one week. An evaluation session had the following structure:

- **Introduction** – participants were given a short tutorial about the interaction process to read (main concept, command list, and generic command usage). The tutorial contained the generic command syntax and one command example (the Tag command). The participants were asked to assume that they were in a realistic setting, in the presence of a BEN supporting system, and to use the command in the example to interact with it for validation.

- **Interactive scenario - command matching exercise** – participants were challenged to complete a matching exercise which consisted on matching a list of visualizations (static images on paper of screenshots) to a list of commands, with a brief functional explanation for each one. The goal behind this was to validate if the command names were found suitable and intuitive for interaction with specific visualizations. Each visualization contained elements that would be associated a specific command use.

- **Command usage and feedback extraction** – participants were inquired regarding command usage for observation and feedback. Each command had a detailed functional description and synopsis. A subset of commands had a possible use scenario based on a screenshot visualization of a public screen. Participants were placed two open-ended questions regarding how they would use the command and what comments or suggestions they had.

5.3.3 OBSERVATIONS

This section presents information obtained from observations during the three phases of the evaluation process.

5.3.3.1 INTRODUCTION

In the introductory part of the session, participants read the tutorial and considered it to be quite clear, feeling no need to place any questions. In order to extract and validate the

user's conceptual model regarding the interaction process, participants were asked to interact using the given command example and to describe the process. Many of the participants revealed initial misconceptions. Despite being clearly stated in what the process of interaction consisted, many of the participants believed that commands were to be sent within messages via Bluetooth push, or that there was some unexplored functionality within their device's configuration menus that would do it.

Interactive scenario - command matching exercise

In the second exercise, participants were given a list of static images and told them to be screenshots of a public screen with BEN support. This visualizations are list in Appendix C.

1. The first image was composed of a header, identifying it as a product evaluation, and three images containing a food product each. Each image was documented with a label, an identifying reference and the average classification of the product.

2. The second image had for header "The most popular politician". It consisted in voting poll were three images of known politicians were each one was associated to a label, a reference, and the respective number of votes.

3. The third image had for header "Emotional States", and consisted on a set of four unequally sized emoticons with a size-proportional number associated to each one. The number indicated the sum of all reported emotional states reported to each category.

4. The fourth image had for header "Choose content to visualize/play", and consisted on a 3 x 3 matrix with 9 items placed in 3 content categories; music, video and presentation. Each one had a name, a reference and an icon according to its type.

5. The fifth image was similar to the previous but had for header "Download content to your handset", and also consisted on a 3 x 3 matrix with 9 items placed in 3 content categories; music, ringtones and wallpapers.

6. The sixth image had for header "Leave your comment", and consisted of a vertical list of comments with associated avatars, made by people in reply to a primal statement made by someone. The initial statement was differentiated by a bigger avatar and had an item reference.

7. The seventh image had no header and consisted of a Tag cloud of musical styles.

8. The eighth image had for header “Leave your message”, and consisted of a vertical list of messages.

9. The ninth image had for header “People around”, and consisted in a vertical list names with an associated avatar.

In this exercise, participants were generally successful in matching command names to visualizations. The first three visualizations, which were expected to be matched with Rate, Vote and MOOD command names respectively, were completed as expected.

In the fourth and fifth visualizations, which were very similar and expected to be matched with SHOW and GET respectively, participants struggled initially but eventually converged to the expected answer. This happened mainly because both visualizations shared a content grid, being the main difference at the header, where different functionality was expressed (download content to handset or play content in the screen).

The sixth visualization was expected to be matched with Comment. In spite of containing word “comment” in the header and having in the first text entry explicitly referenced, a minority of participants matched this visualization with Say. In fact, in the eighth visualization, which was expected to be matched with Say some participants matched it with comment was well.

The seventh visualization which consisted of a Tag cloud of musical styles had less unanimity. The majority the participants matched the Tag command name, one participant matched the name GROUP, and the remaining matched Music.

Regarding the ninth and last visualization, all participants associated the Id command name.

Command usage and feedback extraction

In the third part of the evaluation, each participant was observed and his answers recorded. For each command at least two open-ended questions were placed:

- The first question asked the participant to describe the usage of a certain command.
- The second question asked for comments about the command, its usage and suggestions to improve it

Some of commands had a use scenario in order to help contextualizing its use. Some commands had additional questions in order to evaluate a more specific usage.

Mood

Overall, the majority of the participants had a correct understanding of how to make a correct usage of the Mood command and followed the correct syntax. One exception was observed though; a participant used a reverse smiley (read from right to left), disregarding the proposed set.

Regarding the purpose of the command, several participants felt a lack of context around the use of this command. One participant admitted to use this command only in a game context.

In the comments/suggestions regarding the Mood command, participants used words like "Intuitive", "Easy" and "Effective" to describe its usage.

One participant commented that the use of several emoticons in one command instance, if not contradictory, could help to make a more precise statement of his emotional state. One other participant suggested the smiley alone to be the command, without the need of any name. Two participants commented regarding the domain of the command value, on the fact that aggregation of smileys in categories would kill their specific expression value. One of them went even further by suggesting indexing a compact code to each emoticon and using this alternate form.

Get

For the evaluation of the get command, it was used an image of the second part of the evaluation, a content grid with referenced items as contextualization scenario. In order to further validate the command, three additional questions were placed in order to challenge the user to download specific content of the grid. All users overcome the challenge successfully, using the command with a correct syntax and with the correct references. The majority of users used the command name abbreviation.

In the comments/suggestions regarding the Get command, participants used words like "Simple", "Easy" and "Cool" to describe its usage. Some of the participants commented on the usefulness of this command as alternate way of getting content to the handset without paying for it, instead of the traditional SMS based method. One participant suggested changing the name of the command to "Download" as he felt this would be more appropriate to describe the operation. One other participant suggested the use of multiple references to enable multiple downloads.

Music

In general, the majority of the participants showed a correct understanding of how to make a correct usage of the Music command. The correct syntax was followed, the use of the "*m"* abbreviation was valued and quotes were used on composed values like bands with two-word names.

Participants expressed their preferences in various ways; a single preference like a band, a list of preferences and several types of preferences ranging from style, to band, to artist... One of the participants expressed her musical preference in a curious way; "*music.i.love."moby!!!""*. This showed some disregard by the semantics of the command.

Another user used preceding values as a preference type indicator; "*music.style.rock.artist.madonna*".

Regarding the purpose of the command, several participants felt, once more, a lack of context supporting the usage of the command.

In the comments/suggestions regarding the Mood command, participants used words like "Intuitive" and "Easy" to describe its usage. One of the participants reached the end of the available characters on the name and commented that size limitations of the Bluetooth name would affect a great deal on the use.

Tag

The majority of the participants showed a correct understanding of how to make a correct usage of the Tag command. The correct syntax was followed, the use of the "*t"* abbreviation was valued and quotes were used on multi-word values. The kind of values used was the expected, meaningful to the user only.

Once more, some participants felt a lack of context supporting the usage of the Tag command. One of the participants felt very confused with its use, because she felt that its usage should be directed by a question, otherwise it would be pointless. Oppositely, another participant referred this to be his favourite command, due to its power of expression. Two participants commented that this would be suitable to gather content and information for display.

Participants used words like "Useful" and "Confusing" to describe its usage.

Group

The majority of the participants showed a correct understanding of how to make a correct usage of the Group command. The correct syntax was followed and the use of the "g*"* abbreviation was valued. The absence of quotes on multi-word values was observed, though. One participant disregarded semantics by using the command something else other that a group, "*group.FCP."Olé!!" "*.

The first question in this group asked participants to use the command to expose their football club. Interestingly, several people showed to be fans of the same club and used very different values to show it. For instance, fans of *Futebol Clube do Porto* used the following values: "*Futebol Clube do Porto*", "*FCP*", "*fcporto*", "*porto*". The same for another club: "*Benfica*", "*glorioso*", "*slb*".

Another interesting finding is the kind of values that the users used. The command explanation specified some examples of social groups like football clubs, geographic containers like cities or countries, professions. However users also used band names for groups, acknowledging being members of the fan group, employing companies and online social networks like Hi5, Twitter and Facebook. One of the participants used only online social networks for values. One of other participant even used Facebook alongside with his email address, in an account login association for Facebook.

Participants used words like “Easy” and “Cool” to comment the usage of Group command. One of the participants considered the use of this command on very specific settings, giving the example of political party reunions and campaigns. Another participant referred this to be very useful on a gaming context.

Id

Once again, the majority of the participants revealed a correct understanding of how to make a correct usage of the Id command, using the correct syntax generally speaking. Some participants expressed doubts on what would be their username for services like Facebook, if their email addresses, which they use for login in the service, or their actual username. One of the participants expressed doubts on whether or not to use quotes on email address value. Some people used several services in one command, like “*id.fb.somename.ytb.someother* ”.

Regarding the purpose of the command, participants showed general understanding of the concept.

In the comments/suggestions regarding the Id command, participants used words like “Easy” and “Interesting” to describe its usage. One participant commented that the command name was not very obvious, and suggested the name *login* instead.

Another participant commented that this could be a good way to make new friends by exposing affinities. One participant commented that he would not use this command in any occasion, justifying that we would not like to expose his email address nor his account public information.

Show

For the evaluation of the Play command, just like for the Get command, a content grid with referenced items was used as contextualization scenario. Three additional questions were also placed in order to challenge the user to select and play specific content of the grid on the screen. Some users felt that this command could be easily confused with Get.

All users overcome the challenge successfully, using the command with a correct syntax and with the correct references. Some users used the command name abbreviation. One of the users used a list of values in order create a playlist.

Participants used words like “Simple”, “Not obvious” to describe its usage. A significant part of the participants commented on the command name focusing on its abbreviation. They believed this was not an obvious association, and that the abbreviation wasn’t intuitive. One participant suggested changing the name to *Play* and its abbreviation to “*p*”.

One participant referred that the functionality of the command would be very useful in waiting contexts like the bus stop, the airport terminal, or during trips.

The suggestions induced a change in the final version of the language, as it in accordance with naming guidelines. The command was renamed to Play

Shout

All participants showed a correct understanding of how to make a correct usage of the Shout command and of its purpose. The correct syntax was followed but the use of the abbreviation wasn't very valued. It was observed a particular attention regarding the use of quotes in this command.

Participants used words like "Easy" and "Perfect" to comment the usage of Say command.

Two of the participants considered that there wasn't an obvious association between the command name and its functionality. One of them suggested changing the name to *"Say"* and the abbreviation to "s" to make it more memorisable. This suggestion also induced a change in the final version of the language. The command was renamed to "Say" and the abbreviation to "s", as the possible collision with Shout ceased to exist.

Vote

For the evaluation of the Vote command it was used a voting poll with referenced images of politicians as contextualization scenario. The majority of the participants showed a correct understanding of how to make a correct usage of the Vote command and of its purpose. All users overcome the challenge successfully, using the command with a correct syntax and with the correct references. The majority of users chose to use the command name abbreviation.

Participants used words like "Easy" and "Perfect" to comment the usage of Vote command. The use of the optional operator raised a few issues, however. Some participants questioned the need of the optional operator, and some of them used it in an unorthodox way in order to take votes away from the poll. Several participants suggested changing the order of the values, in order to accept the operator after the reference, arguing to be more intuitive. One participant suggested extending the command to also accept a descriptive label instead of a reference.

Rate

For the evaluation of the Rate command it was followed the same system as previously for Vote. A product showcase with referenced images of food products was used as contextualization scenario. The majority of the participants showed a correct understanding of how to make a correct usage of the Rate command and of its purpose, using the command with a correct syntax and with the correct references. The majority of users also chose to use the command name abbreviation. One participant, although unsure, tried to rate two products with in one instance of command. Another participant used the grade value in form 6/10, expressing the range of the evaluation himself.

Participants used words like "Simple", "Intuitive" and "Functional" to comment the usage of Rate command. Almost all participants commented on the order of the values, arguing that the reference should precede the evaluation grade. One participant suggested extending the command to also accept a descriptive label instead of a reference.

Comment

For the evaluation of the Comment command it was used the Rate command contextualization scenario, to allow participants to direct comments to the product showcase. The majority of the participants showed a correct understanding of how to make a correct usage of the Comment command and of its purpose, using the command with a correct syntax and with the correct references. All users chose to use the command name abbreviation.

Participants used words like “Simple” and “Intuitive” to comment the usage of Rate command. Several participants commented on abbreviation, which was difficult to memorize, “c” would be most intuitive. One participant suggested merging the Id command in order to complement the comment.

Uri

All participants showed a correct understanding of how to make a correct usage of the Uri command. The correct syntax was followed but the use of the abbreviation wasn’t very valued. It was observed a particular attention regarding the use of quotes in this command.

The urls used in the command Uri were either generic like “www.google.com” or “http://www.sapo.pt”, sites of newspapers like “www.publico.pt”, or personal like blogs and personal pages or their company’s home page. Some participants found no utility in this command, arguing that web pages would lose their interactivity. Others found this a precious way to advertise and to share contents.

Participants used words like “Easy” to comment the usage of Uri command. Some participants commented on the need to use the protocol “http://”, and suggested that modern browser capabilities exclude this need, thus this should be considered for the use of this command.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

This work presented a study over Bluetooth naming and introduced BEN as technique for simple interaction around situated displays. Naming guidelines to maximize the efficiency and potential use of BEN were presented as supporting dimensions of the command language design process. Text size limitations, data entry mechanism limitations, and support for known data formats were identified as decisive and were used together with interaction requirements to the specification of the command language. The Instant Places system was presented and BEN integration was contextualized. The outcome of the language design process, the final version of the command language that supports BEN was also presented and was published in http://ubicomp.algoritmi.uminho.pt/ben.

Three user research studies were made in order to support the interaction design process at different phases. The first study consisted on a usability study conducted to obtain preliminary insight on general use of Bluetooth and on procedural mechanisms associated with Bluetooth naming. The main goal in obtaining this knowledge was to define the naming guidelines upon it. Results showed that, regarding the general Bluetooth use, the majority of the interviewed people used the Bluetooth radio turned off, or most occasionally turned on. Sharing content was the main reason pointed to turn Bluetooth on. Regarding naming practices, only a small minority admitted to change the name occasionally or frequently. Regarding the task execution, results showed a good performance either on Bluetooth activation and naming. During this study it was observed the impact of text size limitations and of the characteristics of the proposed syntax, like command separators, abbreviations and command name case. These results validated the use of Bluetooth naming as a possible interaction technique, but also provided material on which to derive naming guidelines.

The second study consisted on a trial deployment of the Instant Places system with support for BEN draft specification that consisted on two commands, Tag and Flickr commands. The main goal behind this study was to measure the adoption of the proposed technique and to obtain some insight regarding the use of the proposed syntax. The overall results of the trial showed a significant adoption of the proposed technique as part of situated interactions around a public display. A small percentage of syntactic misuses was observed, mainly regarding the use of spaces, which was used to inform the command language design.

The third study consisted on a qualitative evaluation of the usage of the final specification of the command language. It was divided in different parts, each defined for the evaluation

of specific properties. The first part intended to validate user's conceptual model after reading a tutorial on the technique. Results showed that there were initial misconceptions regarding the use of Bluetooth naming. It was observed that the traditional use of Bluetooth was somehow present and conditioned the perception of the proposed technique.

The second part of the study consisted on a visualization-command name matching exercise where the goal was to evaluate the intuitiveness of commands. Results showed a successful completion in the general. There were some exceptions observed however, regarding Say-Comment commands and Tag-Group-Music which were given ambiguous uses by some of the participants.

The last part of this study focused on the usage and feedback about the commands. Overall results were positive; in general, a correct understanding and usage of the commands' syntax was observed in the majority of participants, which they described as easy and intuitive. Some issues were systematically verified; the optional operator of the Vote command; the abbreviations of the Show and Comment commands, the lack of contextualization on some of the commands of the self-exposure category. Some of the design considerations were valued, like the abbreviations which were used by a majority of people, and the use of quotes as delimiters of composed arguments. Some suggestions were taken into account to improve the language.

Bluetooth Extended Naming proved to be a functional and effective technique for situated interaction. The potential limitations that were anticipated, regarding public rejection and technical difficulties, revealed little impact on its use. The prototype study indicated it was significantly adopted and valued. However, it is necessary to consider and understand the characteristics of the medium and its main limitations, with particular regard to the text size limitations.

Regarding future work, the usage of this technique can be further explored. A field deployment of Instant Places with full support of the final version of BEN command language would prove to be useful. This would allow the observation of the usage and appropriation of this technique by users in a real setting and how it would be leveraged in social practices.

BIBLIOGRAPHY

[1] Mark Weiser, “The computer for the 21st century,” *SIGMOBILE Mob. Comput. Commun. Rev.* 3, no. 3 (1999): 3-11.

[2] Rui Jose and Francisco Bernardo, “Extended Bluetooth Naming for Empowered Presence and Situated Interaction with Public Displays,” in *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, 2009, 57-65, http://dx.doi.org/10.1007/978-3-540-85867-6_7.

[3] R. Jose et al., “Instant Places: Using Bluetooth for Situated Interaction in Public Displays,” *Pervasive Computing, IEEE* 7, no. 4 (2008): 52-57.

[4] Tim Kindberg and Timothy Jones, ““Merolyn the Phone”: A Study of Bluetooth Naming Practices,” in *UbiComp 2007: Ubiquitous Computing*, 2007, 318-335, http://dx.doi.org/10.1007/978-3-540-74853-3_19.

[5] Eric Paulos and Elizabeth Goodman, “The familiar stranger: anxiety, comfort, and play in public places,” in *CHI '04: Proceedings of the 2004 conference on Human factors in computing systems* (ACM Press, 2004), 230, 223, http://dx.doi.org/10.1145/985692.985721.

[6] Tom Nicolai et al., “Exploring Social Context with the Wireless Rope,” in *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, 2006, 883, 874, http://dx.doi.org/10.1007/11915034_112.

[7] V. Kostakos and E. O’Neill, “Capturing and visualising Bluetooth encounters,” *In adjunct proceedings of the conference on Human factors in computing systems (CHI 2008)* (2008), http://dme.uma.pt/people/faculty/vassilis.kostakos/Publications.html.

[8] Joseph F. McCarthy et al., “Proactive Displays & The Experience UbiComp project,” *SIGGROUP Bull.* 23, no. 3 (2002): 38-41.

[9] Nathan Eagle and Alex Pentland, “Reality Mining: Sensing Complex Social Systems,” *Journal of Personal and Ubiquitous Computing* (2005), http://reality.media.mit.edu/pdfs/realitymining.pdf.

[10] Daantje Derks, Arjan Bos, and Jasper Grumbkow, “Emoticons and social interaction on the Internet: the importance of social context,” *Computers in Human Behavior* 23, no. 1 (January 2007): 849, 842.

[11] “Microformats | About microformats,” http://microformats.org/about/.

[12] “Picoformats · Microformats Wiki,” http://microformats.org/wiki/picoformats.

[13] “Microblogging-nanoformats - Microformats Wiki,” http://microformats.org/wiki/twitter-nanoformats.

[14] ”Specification of the Bluetooth System, Core Version 2.1 + EDR” Bluetooth SIG, 2004.

[15] “Specification of the Bluetooth System, Core Version 3.0 + HS” Bluetooth SIG, 2009.

[16] Myra Dideles, “Bluetooth: a technical overview,” *Crossroads* 9, no. 4 (2003): 11-18.

[17] “Bluetooth.com | History of Bluetooth Technology.” http://www.bluetooth.com/Bluetooth/SIG/History_of_the_SIG.htm

[18] “Bluetooth.com | Communication Topology,” http://bluetooth.com/Bluetooth/Technology/Works/Communications_Topology.htm.

[19] “Bluetooth.com | Architecture - Baseband,” http://bluetooth.com/Bluetooth/Technology/Works/Architecture__Baseband.htm.

[20] “Bluetooth.com | Core System Architecture,” http://bluetooth.com/Bluetooth/Technology/Works/Core_System_Architecture.htm.

[21] Minimizing Bluetooth Interference http://www.wi-fiplanet.com/tutorials/article.php/1379911

[22] “Wi-Fi and Bluetooth: Enabling Cooexistence.” http://www.ce-mag.com/archive/01/05/lansford.html

[23] R.W. Woodings et al., “Rapid heterogeneous ad hoc connection establishment: accelerating Bluetooth inquiry using IrDA,” in *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, vol. 1, 2002, 342-349 vol.1.

[24] B.S. Peterson, R.O. Baldwin, and J.P. Kharoufeh, “Bluetooth Inquiry Time Characterization and Selection,” *Mobile Computing, IEEE Transactions on* 5, no. 9 (2006): 1173-1187.

[25] B. Peterson, R. Baldwin, and R. Raines, “Bluetooth Discovery Time with Multiple Inquirers,” Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06), Kauia, HI, USA: , pp. 232a-232a.

[25] “T9 (predictive text) - Wikipedia, the free encyclopedia,” http://en.wikipedia.org/wiki/T9_(predictive_text).

[26] “Statistics on the Fastest Phone Keyboard Data Entry Method,” http://phonekeyboard.com/Statistics.html.

[27] "RFC 1035 - Domain names - implementation and specification," http://tools.ietf.org/html/rfc1035.

[28] "RFC 5321 - Simple Mail Transfer Protocol," http://tools.ietf.org/html/rfc5321.

[29] "RFC 1738 - Uniform Resource Locators (URL)," http://tools.ietf.org/html/rfc1738.

[30] ISO - International Organization for Standardization, "ISO 4217 - Currency names and code elements," Text, http://www.iso.org/iso/support/currency_codes_list-1.htm.

[31] ISO - International Organization for Standardization, "ISO 9241-15 - Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 15: Command dialogues.," Text, http://www.iso.org/iso/catalogue_detail.htm?csnumber=16887.

[32] "Twitter: What are you doing?," http://twitter.com/.

[33] "Adocu - Nanoblogging," http://adocu.com/.

[34] Jakob Nielsen, *Usability Engineering*, New edition. (Morgan Kaufmann, 1993).

[35] K. Gomoll and A. Nicol, "Guidelines for user observation," http://grouplab.cpsc.ucalgary.ca/saul/hci_topics/assignments/usability/ass2_usabil.html.

[36] "CodeProject: a Tiny Parser Generator v1.2. Free source code and programming help." http://www.codeproject.com/KB/recipes/TinyPG.aspx

[37] "Bluetooth.com | History of Bluetooth Technology." http://www.bluetooth.com/Bluetooth/SIG/History_of_the_SIG.htm

APPENDIX A

APPENDIX A

QUESTIONNAIRE: BLUETOOTH INTERACTION EVALUATION

Questionário de avaliação da interacção com o Bluetooth

Este questionário tem por objectivos estabelecer os perfis gerais de utilização do Bluetooth em dispositivos móveis (**telemóveis**, **PDA**, **etc**), e determinar os condicionalismos existentes nessa utilização. Pede-se que responda às questões colocadas de forma sincera e ojbectiva. As suas respostas são confidenciais e serão apenas objecto de tratamento estatístico. Caso não perceba alguma questão ou não seja possível responder, agradecemos que deixe a resposta em branco.

Idade: _____ **Género**: ○ **M** ○ **F**

Marca e modelo de telemóvel utilizado: _____

Q1. Já alguma vez activou o Bluetooth no seu dispositivo ?

○ Sim

○ Não

Q2. Como costuma utilizar o Bluetooth ?

○ Sempre inactivo ou invisível

○ Ocasionalmente visível

○ Frequentemente visível

○ Sempre visivel

Porquê ? _____

Q3. Qual a principal razão para activar o Bluetooth no seu telemóvel?

○ Curiosidade

○ Pesquisa de outros dispositivos

Q3. Qual a principal razão para activar o Bluetooth no seu telemóvel?

○ Curiosidade

○ Pesquisa de outros dispositivos

○ Emparelhamento com outro dispositivo

○ Troca de conteúdos

○ Outra - Qual? _____

Q4. Com que frequência activa o Bluetooth ?

○ Várias vezes ao dia

○ Frequentemente

○ Ocasionalmente

○ Muito raramente

○ Nunca

Qual a principal razão para isso ? _____

Q5. Acho seguro andar com o Bluetooth activo no meu dispositivo móvel.

Discordo **Concordo**

○ ○ ○ ○ ○ ○ ○ ○ ○ ○

1 2 3 4 5 6 7 8 9 10

Q6. Já atribuiu um nome ao seu dispositivo ?

○ Sim

○ Não

Q7. Com que frequência renomeia o Bluetooth ?

- ○ Várias vezes ao dia
- ○ Frequentemente
- ○ Ocasionalmente
- ○ Muito raramente
- ○ Nunca

Qual a principal razão para isso ? _____

Q8. Qual o número de caracteres que o seu telemóvel suporta para o nome Bluetooth ? (para verificar isto, insira caracteres no nome até não conseguir mais**) _____**

Tarefas

Nesta secção do questionário pretende-se que concretize uma série de tarefas, tendo em conta o tempo que cada uma demora. O objectivo é determinar tempos médios associados à tarefa, bem como dificuldades específicas na sua execução. Pede-se, portanto, que as cumpra a uma velocidade seja lhe seja natural, sem pressas, e que **contabilize o tempo exacto** que demorou.

Grupo I

Neste grupo, pretende-se que cumpra as tarefas na totalidade, e que inicie cada uma delas **a partir do menu default** (*desktop*) do dispositivo, sem fazer uso de atalho ou *speed dial.*

T1. Activar o Bluetooth no telemóvel

a. Tempo necessário para concluir a tarefa: _____

b. Enfrentei dificuldades ao activar o Bluetooth no meu telemóvel.

Discordo **Concordo**

○ ○ ○ ○ ○ ○ ○ ○ ○ ○

1 2 3 4 5 6 7 8 9 10

b. Comentários sobre a tarefa: _____

T2. Atribuir/Alterar o nome do dispositivo Bluetooth

a. Tempo necessário para concluir a tarefa: _____

b. Enfrentei dificuldades a nomear/renomear o meu telemóvel.

Discordo **Concordo**

○ ○ ○ ○ ○ ○ ○ ○ ○ ○

1 2 3 4 5 6 7 8 9 10

GRUPO II

Neste grupo, pretende-se que cumpra as tarefas na totalidade, e que inicie cada uma **a partir do menu de configuração de Bluetooth**, adicionando ao nome já existente.

T3. Introduzir no nome do dispositivo Bluetooth:

1 comando (CMD) c/ 1 valor (activar)

(ex: johnie CMD:activar)

a. Tempo necessário para concluir a tarefa: _____

b. Comentários sobre a tarefa: _____

T4. Introduzir no nome do dispositivo Bluetooth:

1 comando (CMD) c/ 2 valores (activar,num1) (ex: johnie CMD:activar,num1)

a. Tempo necessário para concluir a tarefa: _____

b. Comentários sobre a tarefa: _____

T5. Introduzir no nome do dispositivo Bluetooth:

- 2 comandos c/ 2 valores (ex: johnie CMD1:activar CMD2:pronto)

a. Tempo necessário para concluir a tarefa: _____

b. Comentários sobre a tarefa: _____

GRUPO III

Neste grupo, pretende-se que concretize **apenas uma** das seguintes tarefas, e que a inicie **a partir do menu de configuração de Bluetooth.**

Para cada uma das tarefas, são exemplificados três formas alternativas de um comando específico (por ex: “tag:”, “tg:”, “t:”), da qual deve ser escolhida uma. Pede-se que justifique a escolha do comando.

T6. Introduzir no nome do dispositivo Bluetooth:

- Comandos ‘*tag*’ com valor (exs: johnie tag:happyToday,

johnie tg:happyToday, johnie t:happyToday)

a. Tempo necessário para concluir a tarefa: _____

b. Comentários sobre a tarefa: _____

T7. Introduzir no nome do dispositivo Bluetooth:

- Comandos ‘*flickr*’ com valor (exs: johnie flickr:party

johnie flk:party, johnie f:party)

a. Tempo necessário para concluir a tarefa: _____

b. Comentários sobre a tarefa: _____

T8. Introduzir no nome do dispositivo Bluetooth:

- Comandos ‘*lastfm*’ com valor (exs: antonio lastfm:radiohead

antonio lfm:radiohead
antonio l:radiohead)

a. Tempo necessário para concluir a tarefa: _____

b. Comentários sobre a tarefa: _____

Obrigado pela colaboração.

APPENDIX B

QUESTIONNAIRE: QUALITATIVE EVALUATION ON BEN COMMAND USAGE

QUESTIONÁRIO DE AVALIAÇÃO DA UTILIZAÇÃO DA B.E.N.

LINGUAGEM PARA INTERACÇÃO COM ECRÃS PÚBLICOS ATRAVÉS DE TELEMÓVEIS COM BLUETOOTH

Este questionário tem por objectivo a recolha de informação relativa à utilização de uma linguagem de comandos simples para interacção com ecrãs públicos.

Pretende-se estudar a utilização espontânea de cada comando definido na linguagem.

Não é necessária muita experiência, apenas saber editar o nome Bluetooth do seu telemóvel.

Leia com atenção, por favor

Considere que no centro comercial ou no bar que costuma frequentar, existe um **ecrã com o qual pode interagir**, através de **comandos que pode escrever no nome Bluetooth do seu telemóvel**. O sistema consegue, portanto, "ler e perceber" esses comandos do nome do seu dispositvo.

Cada comando pode ser utilizado **isoladamente** ou **combinado com outros comandos**, desde que haja disponibilidade de caracteres no nome do dispositivo.

APPENDIX B

A forma genérica de um comando é a seguinte,

Nome_do_comando **.** *valor* **.** *valor* **.** *valor* **.** *valor* **,**

Isto é, o **nome do comando** seguido de um **ponto**, seguido de **um ou mais valores** também **separados por ponto**. Se um valor for composto por **várias palavras**, deve ser **delimitado por aspas.** Por exemplo, para o comando **Tag**,

Tag **.**festa **.**férias **.**"Party time"

Considere que existe um comando **MOOD**.

Este comando serve para expressar estados emocionais ou expressões faciais ao sistema, através de ícones ou *smileys* (tabela em baixo).

Alegre	Triste	Surpreso	Zangado	Segredo
:) :-) :-D |-) |-D :->	: (:- < :-c :'-(:-| :-[	:Q :-o 8-O :O 8-|	>:-|):-@):-V >:-<	:-, :-# :-X :-Y :-"
Irónico	**Chateado**	**Brincalhão**	**Confuso**	**Apaixonado**
:-, :-P :-r	:-\ :-I :-/ |-{	;-) ;-> :*) :-T	:-& :-S :-?	: * E> <3 S2

Para utilizar este comando, deve seguir a forma,

mood .*smiley*

Questões:

APPENDIX B

Considere que o ecrã lhe apresenta a seguinte vizualização com a qual pode interagir:

1. Descreva como utilizaria o comando **mood** para interagir com o sistema.
2. Comentários/sugestões sobre a utilização do comando **mood**

++

Considere que existe um comando **get.**

Este comando serve para fazer o download para o telemóvel de conteúdos (videos, imagens, música, ringtones) que são mostrados no ecrã. Cada item tem associada uma referência ou etiqueta, que é para ser utilizada no comando **GET** para o efeito.

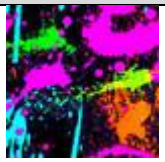
Após algum tempo o sistema injecta o conteúdo indicado no seu telemóvel.

Para utilizar este comando, deve seguir uma das formas,

get . *referência_do_conteúdo* ou **g .** *referência_do_conteúdo*

Questões:

Considere o seguinte cenário. O ecrã apresenta-lhe uma grelha, com conteúdos que pode descarregar para o seu telemóvel através da respectiva referência.

Descarregue conteúdos para telemóvel			
Música/ MP3	 **"Eu tenho 2 amores"**	 **"Gente da Minha Terra"**	 **"Como o macaco gosta de banana"**
	Ref: **m001**	Ref: **m002**	Ref: **m003**
Toque polifónicos	**"Only you"**	**"Satisfaction"**	**"Hey Mr DJ"**
	Ref: **t101**	Ref: **t102**	Ref: **t103**
Imagens/ Wallpapers			
	Ref: **i201**	Ref: **i202**	Ref: **i203**

Utilizando o comando "**Get**" descreva como faria para descarregar uma **música** para o seu telemóvel

1. Utilizando o comando "**Get**" descreva como faria para descarregar um **toque polifónico** para o seu telemóvel

2. Utilizando o comando "**Get**" descreva como faria para descarregar um **toque imagem** para o seu telemóvel

3. Comentários/sugestões sobre a utilização do comando **get**

++

Considere que existe o comando **music.**

Este comando serve para expressar ao sistema preferências musicais como:

- estilos musicais (ex: rock, pop, jazz)
- artistas (ex:Madonna, Michael Jackson)
- bandas (ex:Metallica, Beatles)
- canções (ex: , Thriller)
- albums
- ...

Para utilizar este comando, deve seguir uma das formas,

music . *valor* **.** *valor ...* ou **m .** *valor* **.** *valor.*

Questões:

1. Mostre como utilizaria o comando **music** para exprimir as suas preferências musicais. ao sistema.
2. Comentários/sugestões sobre a utilização do comando **music**

Considere que existe um comando **"Tag".**

Este comando serve para indicar ao sistema uma **palavra-chave** que, de um modo informal e subjectivo, identifique algo.

Para utilizar este comando, deve seguir uma das formas,

Tag . *palavra-chave* **.** *palavra-chave* **. .**...

ou

t . *palavra-chave* **.** *palavra-chave* **.** ...

Questões:

1. Descreva como usaria o comando **Tag** para interagir com o sistema
2. Comentários/sugestões sobre a utilização do comando **Tag**

++

Considere que existe um comando **"group"**.

Este comando serve para indicar ao sistema palavras-chave que, de um modo informal, identifiquem, **grupos sociais** a que o utilizador pertence, como clube de futebol, cidade, profissão, curso.

Para utilizar este comando, deve seguir uma das formas,

group . *grupo* **.** *grupo* **.** *...*

ou

g . *grupo* **.** *grupo* **.** *...*

Questões:

1. Descreva como usar o comando **group** para indicar o clube de futebol a que pertence.
2. Descreva como usar o comando **group** para indicar os grupos sociais a que pertence.
3. Comentários/sugestões sobre a utilização do comando **group**

Considere que existe um comando **"id"**.

Este comando requer que o utilizador seja membro de sites comunitários ou redes sociais on-line como o Facebook, LastFm, Flickr ou YouTube.

A sua função é que o utilizador se **identifique** e se **exponha** perante o sistema através do seu nome de utilizador. Deste modo, está a ceder ao sistema informação pública do seu perfil ou conta dos serviços indicados.

Para utilizar este comando, deve seguir a forma,

id . *serviço* **.** *username*

onde o *serviço* poderá tomar uma das seguintes formas

Nome de serviço	Abreviatura
facebook	fb
flickr	flk
lastFM	lfm
youTube	ytb
technorati	tch
eventful	evt
twitter	twt
linkedIn	lki

Questões:

1. Descreva como indicaria ao sistema, através do comando **id** o seu perfil ou conta pessoal de um serviço ou rede social a que pertence.
2. Comentários/sugestões sobre a utilização do comando **id**

Considere que existe um comando **"show".**

Este comando serve para fazer *play* a conteúdos (como videos, imagens, apresentações) que são mostrados no ecrã e aos quais está associada uma referência ou identificador, para este efeito.

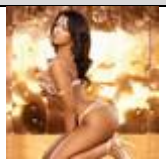
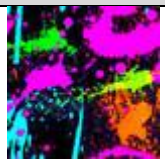
Para utilizar este comando, deve seguir uma das formas,

show . *referência_do_conteúdo* ou **shw .** *referência_do_conteúdo*

ou seja, deve inserir **o nome do comando** (ou abreviatura) seguido da **referência** do conteúdo, **separados por um ponto**.

Questões:

Considere o seguinte cenário. O ecrã mostra-lhe uma grelha com conteúdos que podem ser visualizados a pedido.

Escolha um conteúdo para visualizar/passar			
Música	 **"Eu tenho 2 amores"**	 **"Gente da Minha Terra"**	 **"Como o macaco gosta de banana"**
	Ref: **m001**	Ref: **m002**	Ref: **m003**
Apresentação Powerpoint©	***Miss60*** Catálogo de moda	**Ementa do Dia**	**Piadas do Sem-Pescoço**
	Ref: **a101**	Ref: **a102**	Ref: a**102**
Video			
	Ref: **v201**	Ref: **v202**	Ref: **v203**

1. Utilizando o comando s**how** descreva como procederia para escolher e ouvir uma **música** da grelha
2. Utilizando o comando s**how** descreva como procederia para escolher e ver um **video** da grelha
3. Utilizando o comando s**how** descreva como procederia para escolher e iniciar uma **apresentação** da grelha
4. Comentários/sugestões sobre a utilização do comando **show**

++

Considere que existe um comando **“shout”.**

Este comando serve para deixar uma mensagem no ecrã, de foma semelhante ao Twitter. Para utilizar este comando, deve seguir uma das formas,

shout . “ *mensagem* **”** ou **sht . “** *mensagem* **”**

Questões:

1. Descreva como usaria o comando **shout** para interagir com o sistema
2. Comentários/sugestões sobre a utilização do comando **shout**

Considere que existe um comando **vote.**

Este comando serve para efectuar o voto, numa *poll* de votações sugeridas pelo ecrã.

As opções de voto são identificadas por uma etiqueta ou têm uma referência associada.

Para utilizar este comando, deve seguir uma das formas,

vote . *operador* **.** *referência_do_conteúdo*
ou
vt . *operador* **.** *referência_do_conteúdo*

onde o *operador* é **opcional**, podendo ser um dos seguintes: '**+**' ou '-'

Questões:

Considere que o ecrã lhe apresenta a seguinte vizualização com a qual pode pode interagir:

1. Descreva como utilizaria o comando **vote** para interagir com o sistema
2. Comentários/sugestões sobre a utilização do comando **vote**

Considere que existe um comando **rate.**

Este comando serve para fazer uma avaliação informal sobre items mostrados pelo ecrã (um filme, o serviço da loja da esquina, o hamburguer que acabei de comer). A avaliação é baseada numa escala numérica, [1-10] e o item a avaliar tem um nome ou uma referência associada.

Para utilizar este comando, deve seguir uma das formas,

rate . [1-10] . " referência/nome "
ou
rt . [1-10] . " referência/nome "

ou seja, deve inserir o **nome do comando** (ou abreviatura) seguido de um **ponto**, seguido da **nota ou valor atribuído**, seguida da **referência** ao item a avaliar por aspas.

Questões:

Considere que o ecrã lhe apresenta a seguinte vizualização com a qual pode pode interagir:

1. Descreva como utilizaria o comando **rate** para interagir com o sistema
2. Comentários/sugestões sobre a utilização do comando **rate**

Considere que existe um comando **"comment".**

Este comando serve para introduzir no ecrã um comentário textual sobre um item específico apresentado.

Para utilizar este comando, deve seguir uma das formas,

cmt . *referência* . " *mensagem* ***"***

Questões:

1. Descreva como utilizaria o comando **comment** para interagir com o sistema
2. Comentários/sugestões sobre a utilização do comando **comment**

++

Considere que existe um comando **uri.**

Este comando serve para introduzir no ecrã um **endereço web** ou um recurso web como uma **página**, uma **imagem**, um **video,** etc.

Para utilizar este comando, deve seguir a forma,

uri . " *http://....* ***"***

Questões:

1. Descreva como utilizaria o comando **uri** para interagir com o sistema
2. Comentários/sugestões sobre a utilização do comando **uri**

APPENDIX C

IMAGE VISUALIZATIONS FROM THE QUALITATIVE EVALUATION INTERVIEW

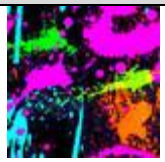
Considere que pode interagir um ecrã publico que vai mostrando cada uma das seguintes visualizações. Dada a listagem de comandos, indique que como utilizaria para interagir com cada uma delas.

Vis.1

Vis.2

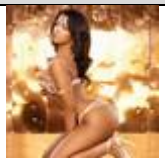
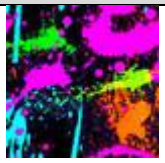
APPENDIX C

Vis.3

Vis. 3Escolha um conteúdo para visualizar/passar			
Música	 **"Eu tenho 2 amores"**	 **"Gente da Minha Terra"**	 **"Como o macaco gosta de banana"**
	Ref: **m001**	Ref: **m002**	Ref: **m003**
Apresentação Powerpoint©	***Miss60*** Catálogo de moda	**Ementa do Dia**	**Piadas do Sem-Pescoço**
	Ref: **a101**	Ref: **a102**	Ref: a**102**
Video			
	Ref: **v201**	Ref: **v202**	Ref: **v203**

Vis.4

APPENDIX C

Descarregue conteúdos para telemóvel			
Música/ MP3	 **"Eu tenho 2 amores"**	 **"Gente da Minha Terra"**	 **"Como o macaco gosta de banana"**
	Ref: **m001**	Ref: **m002**	Ref: **m003**
Toque polifónicos	**"Only you"**	**"Satisfaction"**	**"Hey Mr DJ"**
	Ref: **t101**	Ref: **t102**	Ref: **t103**
Imagens/ Wallpapers			
	Ref: **i201**	Ref: **i202**	Ref: **i203**

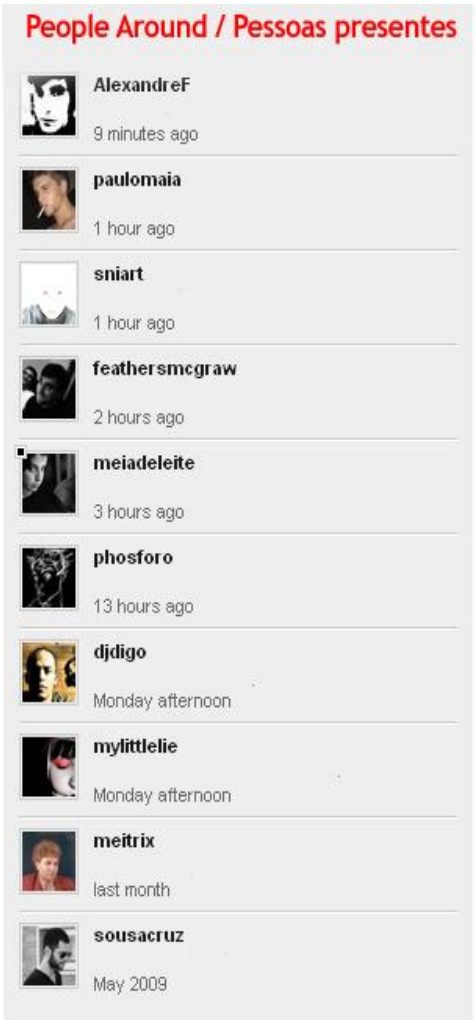
Vis. 5

Vis. 6

APPENDIX C

Vis. 7

Vis.8



Vis. 9