Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

MSCC Dissertation in Computer Engineering
1st Semester, 2008/2009

Developing Persuasive Interfaces

Pedro Jorge Gonçalves Lobo

Supervisor: Prof. Teresa Romão

Lisbon

2009

Number of the student: 26509

Name: Pedro Jorge Gonçalves Lobo

Title:

Developing for Persuasive Interfaces

Keywords:

Attitude and behaviour change

Persuasive computing

Ubiquitous computing

Acknowledgments

First and foremost, I would like to thank Professor Teresa Romão for her supervising on the elaboration of this master thesis. I could not forget to express my gratitude to Ydreams, in particular to José Carlos Danado and A. Eduardo Dias for their good advices and to David Palma for providing the sensors that allowed developing part of this work. Also, I would like to thank to all my family, friends and colleagues who helped me with their support in various ways, especially to Laetitia Mendes for her help with children's tests and to Miguel Ângelo who provided the means for the realization of these same tests.

Resumo

Actualmente, os computadores são ferramentas indispensáveis para a maioria das pessoas. Dado que os computadores são desenvolvidos maioritariamente para facilitar as tarefas das pessoas, os sistemas deverão ser pensados e construídos com o objectivo de se tornarem cada vez mais úteis. Hoje em dia as pessoas ainda têm de passar por um processo de aprendizagem antes de conseguirem utilizar uma dada aplicação, mas espera-se que no futuro este processo não exista. Idealmente a tecnologia chegará a um ponto em que estará tão indetectável no ambiente que interagir com ela se torna natural, tal como com qualquer outro objecto.

Os computadores têm vindo a tornar-se parte activa do nosso dia-a-dia e tornaram-se indispensáveis na realização de grande parte das nossas tarefas. Esta tese de mestrado tem como objectivo estudar os computadores como tecnologia persuasiva. Na construção de sistemas persuasivos é importante considerar que as pessoas são únicas e diferentes umas das outras. É, então, intuitivo que quanto mais adaptado um sistema estiver às pessoas com quem está a interagir, menos intrusiva e mais agradável se tornará a interacção. Perante esta ideia, é essencial que o sistema consiga determinar o máximo de aspectos do meio ambiente que consiga.

De forma a alcançar uma boa solução para este conceito, foi realizado um estudo sobre sistemas persuasivos, que é apresentado neste documento. Foi desenhada uma arquitectura para fornecer uma infra-estrutura para o desenvolvimento de aplicações persuasivas. Foi também desenvolvida uma *Authoring Tool* que tem como objectivo permitir a criação de ambientes persuasivos a utilizadores sem conhecimentos de programação.

Além disto, foi também desenvolvido um protótipo de uma aplicação persuasiva como caso de estudo. Foram realizados testes para estudar a sua usabilidade de forma a perceber se o resultado pretendido para o protótipo, *Smart Bins*, fora obtido. Foram feitos outros testes de utilizadores com crianças, o público-alvo da aplicação, para avaliar a sua usabilidade e poder de persuasão. Uma descrição do *Smart Bins*, bem como os resultados desses testes são também apresentados nesta tese. Os resultados proporcionam uma melhor compreensão e novas perspectivas quanto ao uso de tecnologia persuasiva, nomeadamente para questões importantes como a preservação do meio-ambiente.

Abstract

Nowadays, computers are indispensable tools for most people. Since computers' role is to make everyone's life easier, systems can be built to be of even more assistance. While today people have to learn how to interact with computers, in the future, instead of having to do that, computer systems will be blended in our everyday things. It is expected that technology will come to a point where it is natural to interact with it like with every other object.

Computers have been gaining their space in our lives being considered machines that can do almost anything. This master thesis has the purpose of studying how computers can be used as persuasive technology. To build persuasive systems it is important to know that humans are very different from each other. Given this idea it is of extreme importance that the system is able to determine as much aspects of the surrounding environment as it can, to process that information and interact with people accordingly.

To achieve a solution that is effective for this concept, a study of persuasive systems is presented in this document. An architecture was developed in order to provide an infrastructure for the development of persuasive applications. An authoring tool was also implemented in order to allow the creation of context aware persuasive applications by users without programming skills.

Furthermore, a persuasive application prototype was developed as proof of concept for this study. Usability tests were performed to analyze if users could successfully create the application prototype, named Smart Bins, using the proposed framework. Further user tests were executed with children, the target users of the application prototype, to evaluate its usability and persuasiveness. A description of the Smart Bins application as well as the results of the user tests are presented in this thesis. The results provide further understanding and new perspectives regarding the use of persuasive technologies as interactive computer products, designed to change people's habits, therefore helping to improve human's attitudes and behaviours towards important matters like environmental preservation.

Index

1. Introduction 1
1.1. General Introduction and Motivation 1
1.2. Description and Context 2
1.3. Presented Solution 3
1.4. Main Expected Contributions 4
1.5 Structure of the Thesis 4
2. State of the Art 7
2.1. Persuasion 7
2.2. Computers as Persuasive Systems 8
2.2.1. Motivating Users to Modify their Behaviour 9
2.2.2. Social Awareness and Support 13
2.2.3 Credibility in Persuasive Systems 15
2.3. Ubiquitous Systems 16
2.3.1 User Aware Environments 17
2.3.2. Environment's Context Detection 20
2.3.3. Persuasion in Ubiquitous Systems 22
2.4 Human-Computer Interaction 24
2.4.1 Human-Computer Interaction Goals 24
2.4.1 Interaction Design Principles 25
2.5 Child-Computer Interaction 27

2.5.1 Motivation and Goals 27
3. Methodology and framework description 31
3.1 Framework Design Methodology 32
3.1.1 Framework Engine Design 34
3.1.2 Framework Application Design 35
3.1.3 Framework Class Diagram 37
4. Framework Implementation and prototype development 39
4.1 Framework Implementation 39
4.1.1 Engine Implementation 40
4.1.1.1 Sensor Processor 41
4.1.1.2 Camera Processor 41
4.1.1.3 Context Processor 42
4.1.2 Authoring Tool Implementation 42
4.2 Prototype Development 50
5. Evaluation and User Studies 53
5.1 Evaluation with User Participation 53
5.2 Framework Studies 56
5.3 Smart Bins Studies 60
5.3.1 Participants 61
5.3.2 Questionnaire 62
5.3.3 Results 62
6. Conclusions and Future Work 69
References 73
Annex A - Smart Bins Creation 77

Index of Figures and Tables

Figure 2.1 – Persuasion Types....8

Figure 2.2 – Behavioural effects in Persuasive Mirror project....11

Figure 2.3 – Clean Sink showing indicator (left), RFID reader (middle) and sink (right)....12

Figure 2.4 – Calorie tracker architecture....18

Figure 2.5 – Health Advisor runtime Architecture....19

Figure 2.6 – Public to Personal Interaction Model....20

Figure 2.7 – “Gossip Wall” Interaction Model....22

Figure 2.8 - Design Cycle….26

Figure 2.9 – Boehm’s spiral model….26

Figure 2.10 - Flow Zone factors….28

Figure 2.11 - Different players have different Flow Zones….28

Figure 2.12 - Designers adapt players’ flow experience through the choices they deliberately build into the experience….28

Figure 3.1 - Application Development process….33

Figure 3.2 - Framework Engine Architecture….34

Figure 3.3 – Elements evolution in the framework….35

Figure 3.4 – Design Model for an Application….36

Figure 3.5 – Framework’s Class Diagram….37

Figure 4.1 - Application's General Settings….43

Figure 4.2 – Application’s General Settings XML Schema….44

Figure 4.3 - Frame edition….45

Figure 4.6 - Frame Settings XML Schema….47

Figure 4.4 – Data Matrix list edition menu….48

Figure 4.5 – Data Matrix lists XML Schema….49

Figure 4.7 – Application’s Structure Preview….50

Figure 4.8 - Smart Bins stand-by menu….52

Figure 5.1 – Test results….59

Figure 5.2 – Users that stood on the right place..63

Figure 5.3– Users that identified card on the camera..63

Figure 5.4 – Users that identified the objects...63

Figure 5.5 – Users inserting objects in the containers..63

Figure 5.6 – Smart Bins Environment……………………………………………………….....65

Figure 5.7 – Users that stood on the right place (2nd day)..66

Figure 5.8 – Users that identified card on camera (2nd day)...66

Figure 5.9 – Users that correctly identified objects with the camera (2nd day)......................................67

Figure 5.10 – Users that inserted the objects in the container (2nd day)..67

Table 2.1 – Computers as Social Actors characteristics...13

Table 5.1 - Classification of experimental and query evaluation techniques………………………….53

Table 5.2 - Classification of observational evaluation techniques…………………………………….54

Table 5.3 – Participants tasks performance results………………………………………………….58

1. Introduction

1.1. General Introduction and Motivation

While computers are being developed at a hallucinating rhythm, its use is frequent at almost every time and place. With technology being as advanced as it is today, there is the possibility to use it to almost anything. The general idea of this work is to study how technology can be used to change people's attitudes and behaviours, to help themselves, the others or the surrounding environment, using persuasive systems. The concept of having computers as persuasive technology to change people's attitudes and behaviours is a very invigorating idea. People are getting accustomed to interact with new technology every day and it is now possible to use this relation between humans and computers to improve human´s behaviours.

The possibility of having computer systems interacting with people, detecting their characteristics and the ambient context allows creating adaptive software that fits people's needs. Moreover, being blended on everyday things in the environment and at the same time not needing conventional user inputs brings another reality to Human-Computer interaction. People are allowed to interact with computer systems as they do with everyday things speeding up the learning process and making it simpler. Also, presenting messages at the right time and place can persuade people to adopt or change ideals and therefore changing their attitudes and behaviours.

This work intends to study how computer systems can be used as persuasive technology to change people attitudes and behaviour towards a sustainable environment. An architecture is proposed and has been developed, in order to provide an infrastructure for the construction of persuasive applications. Furthermore, a prototype was developed as proof of concept for this study. The prototype relates with the use of persuasive interfaces to promote recycling of waste materials, preventing greater future damages in the environment. The results provide further understanding and new perspectives for the use of persuasive technologies as interactive computer products, designed to change what people believe and do, helping to improve human's attitudes and behaviours. The motivation to develop this master thesis comes as a following idea of the problem presented above. Being concerned about the environmental problems that ruin our planet, the key idea of this work is to use technology to provide a system that allows the possibility to persuade people to change their behaviour.

1.2. Description and Context

This work has to do with the study of technology use to help people changing their attitudes and behaviours. This intention comes with the fact that sometimes people do not understand the problems of their inappropriate behaviours and they need to be reminded of that to change their current ideals and opinions that lead to different behaviour's adoption.

Persuade persons to change their behaviour can be a very complex task, since the way people react is not linear as it has to do mostly with human psychology. People do not respond to the same stimuli, as everyone is different from each other. Nonetheless, the system has to be attractive and non-intrusive, capturing the characteristics of the surrounding environment, detecting user's attitudes and responding accordingly. The challenge with interactive and persuasive systems is to understand how to create a model of interaction that captivates everyone who uses the system. One of the problems presented is the choice of proper messages (text, images, video, animations or audio) to provide, considering the person who is interacting with the system, and the right time to present them the information. Usually, people do not like being told what to do and that has to be taken in consideration. Besides, people will only change their behaviours if they feel that the information is credible. Given these facts, all the content displayed to the users has to be credible. The system should provide to users an approach to make them think about their current actions so they realize that they are not acting the way they should and they make the effort to change their behaviours.

As described in section 1.1, the proof of concept focuses on the issue of recycling. The idea is to build a prototype application that draws people's attention for the need of recycling and teach them good recycling practices. Users must feel that they can help to make a difference avoiding the degradation of the environment. To achieve this, one must provide the messages and stimuli that can influence them to form new ideals, which will lead them to adopt new positive behaviours.

This master thesis is related with the project SEEK (Strategic Experiences for Environmental Knowledge), whose first developments were published in Persuasive 2007 [1]. Previous (and on-going) work has been done in the scope of Joana Hipólito's PhD thesis related with EcoDesign.

1.3. Presented Solution

The approach for the solution to the problem presented above involves the creation of a framework that allows the development of new persuasive ubiquitous environments. To change people's behaviours, the system must be aware of the context in question (user, environment and situation) to be able to propose appropriate actions and guide the users through an entertaining and motivational experience.

The framework is supported by sensors to detect the context and users in the system's surroundings. This way, the type of the messages to present and the interaction model can be chosen within that context. An authoring tool was also implemented in order to allow the creation of context aware persuasive applications to users without programming skills. Furthermore, a prototype application was developed as proof of concept in order to determine a possible result of the persuasion effects of this application developed with the use of this framework.

The prototype, Smart Bins, consists in an application to change people's behaviours towards recycling. The intention is to provide social and environmental conscience, especially on children, so they can help to improve the environment in the future. The system detects user's presence, starting the interaction from that moment on and providing an interaction model to teach and motivate children to recycle waste materials. During all the interaction, the system detects the ambient context and presents stimuli according to user interaction. Smart Bins assigns simple tasks for users to follow, in order to create a challenge to carry out and to stimulate them during the interaction. Furthermore, the application provides different stimuli to the users according to the objectives that they reach.

The authoring tool was created to support the architecture and the prototype application was created using the developed system. This application was tested by target users in order to evaluate its usefulness and usability.

1.4. Main Expected Contributions

This is expected to contribute to the development of new persuasive systems. This study and implementation provide a platform to develop new ubiquitous systems that help to improve people's behaviours. The developed authoring tool allows the creation of ubiquitous persuasive applications to users without programming skills. It is also expected that this work provides a new vision of computer systems embedded in everyday objects, not only as tools to help people in their work, but also as entertainment and educational tools that could help users to adopt new positive behaviours.

With the application prototype, the ambition is to raise social conscience and environmental concerns, reducing pollution in order to obtain a healthier world. Furthermore, it is also relevant to study how people react to ubiquitous technology. The use of ubiquity is intended to provide unobtrusiveness between people and computer systems, so they can focus more on the message and tasks and less on learning how to use the system.

The results provide further understanding and new perspectives for the use of persuasive technology, as interactive computer products designed to change what people believe and do, helping to improve human's attitudes and behaviours.

1.5 Structure of the Thesis

The remainder of this thesis is structured as follows.

Chapter 2 ("State of the Art") is related with the background study of this work. Several related topics were studied and are presented in order to contextualize the followed approach and present this thesis related work. In Chapter 3 ("Methodology and framework description") one introduces the methods followed and presents the current solution. In Chapter 4 ("Framework Implementation and prototype development"), the implementation of the framework and the proof concept application are analysed and described in detail.

Chapter 5 (“Evaluation and User Studies”) describes the evaluation methods and the user tests performed for both the authoring tool and the application prototype. Furthermore, the results of these tests are analysed and discussed in this chapter. Finally, in Chapter 6 (“Conclusions and Future Work”) conclusions are presented, followed by some orientations for future improvements.

2. State of the Art

2.1. Persuasion

Not everyone agrees on what persuasion exactly means. **"**Persuasion is a form of social influence. It is the process of guiding people toward the adoption of an idea, attitude, or action by rational and symbolic (though not always logical) means."[1] Fogg [2] defines persuasion as an attempt to change attitudes or behaviours (or both) without using coercion or deception. Persuasion tries to modify the way others think, feel, or act. Persuasion can be seen everywhere: at home, in the streets or at work. For instance, when parents encourage their children to develop better study habits or when politicians try to convince electors to vote for them. Everyone uses persuasion, intentionally or non-intentionally, more or less successfully. Some persuasion techniques are more efficient than others depending on the target users and the contextual situations.

Persuasion can be used to change people's behaviours. For example, stimulating people with inappropriate behaviours, such as drinking, overreacting, smoking, driving recklessly, to change their habits and adopt beneficial behaviours. Persuasion can be applied in several contexts, including commerce, health, education, environment and safety. However, most applications of persuasive techniques are in the field of marketing and advertising.

To Harjumaa et al. [3] there are three types of persuasion: interpersonal persuasion, computer-mediated persuasion, and human-computer persuasion as shown in Figure 2.

[1] Hhttp://en.wikipedia.org/wiki/Persuasion

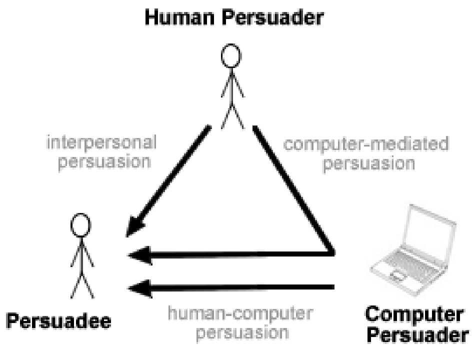

Figure 2.1 – Persuasion Types

Interpersonal persuasion happens when a person persuades other using verbal or non-verbal communication. Computer-mediated persuasion occurs when someone uses technology to persuade a person, through e-mails for example. Human-computer persuasion, that is this study's focus, is the use of persuasion through human-computer interaction.

2.2. Computers as Persuasive Systems

"*Simply put, a persuasive computer is an interactive technology that changes person's attitudes or behaviors.*" [4]. In short, this means that a computational system can be used to change user's ideas and habits. The difference between the two concepts given above, attitude and behaviour change, is the fact that attitude change is accomplished if one can change what people plan to do in a specific moment and behaviour change will only be consummated if someone changes his or her way to think about some subject, making the attitude become an habit.

The study of computers as persuasive technology is relatively new. Fogg [4] coined the term Captology, an acronym for the sentence "computers as persuasive technologies". As stated by Fogg, Captology focus on the design, research, and analysis of interactive computing products created for the purpose of changing people's attitudes and behaviours. In his book "Persuasive Technology" [2], he describes this research and presents several application examples.

As this work focuses on persuasion through Human-Computer interaction, one has to compare computers and people as persuaders. According to Fogg [2], when it comes to persuasion, computers have many advantages over humans:

- Are more persistent. Computers can repeat the same task over and over again.

- Have multiple output forms, such as graphics, videos and audio. Often people are influenced not only by the information itself, but by how it is presented.

- Store, access and manipulate great volume of data.

- Allow anonymity. People can interact with a system without anyone knowing.

- Can be ubiquitous. Computers may even be in places that it is strange for people to be, like bathrooms or bedrooms, which allows them to intervene at the right time and place.

- Are easily scalable. If a system is good it can be reproduced; with humans it is not that easy as it involves training and skills.

Given the advantages of computers over humans concerning persuasion activities, the question in hand is how to have people changing their attitudes and behaviours.

2.2.1. Motivating Users to Modify their Behaviour

There are many ways to address people and try to convince them to do things, some more efficient than others. People can also be mislead, being convinced to do something with the expectation that it will help them and get disappointed at the end. Coercion can even be used to force people to change their behaviour, but persuasion, in the context of persuasive technology, should imply voluntary changes. Nevertheless, the border between coercion and persuasion is not very clear.

It is important to state that when dealing with people, human psychology should be taken in consideration. A significant aspect to have in mind is people receptivity to the system. People tend to dislike intrusive systems that are always telling them what to do. On the other hand, if after interacting with a computer, people realize that the consequences of their attitudes may help them or the community, then there is a great possibility that they will change their attitude and possibly their behaviour, sometimes being persuaded to change without even noticing. Given this fact, one has to think about appealing and non intrusive interfaces that can make everyone realize the benefits in changing their attitudes and behaviour, creating new ideals for people to follow on time to come.

In his book, B.J. Fogg refers that there are many aspects to deem when using computational systems as persuasive tools. The concept of computers as tools comes with the fact that computers may provide ways to simplify user's tasks and interaction according to people's needs and system's goals. Reduction, tunnelling, tailoring, suggestion, self-monitoring, surveillance and conditioning are the tools enunciated [2].

Humans always look for a way to simplify tasks. Using computational power will allow complexity reduction on user's tasks to simplify the process, leaving the rest of the work to the computer. This is what reduction is all about.

The idea underlying tunnelling corresponds to the use of technology to guide a user through an experience following a predetermined sequence of actions, being able to use persuasion along the way. This means that while users interact with a system it can show them messages or content that suggests the user what to do, providing them the information that hopefully will make them change their minds. Also, even when the computer is performing other tasks, it still should be able to provide information to the users so they always keep their attention on the displayed contents. This way, no time will be wasted and users are always being provided with useful information.

People should sense that the information is tailored for them, which means that, for instance, if the interface is designed for different individuals, it should have multiple forms of display and interaction according to individual's needs, interests, personality or usage context. Figure 3 shows a representative example of a persuasive technology project by A. Andrés et al. [5] using tailored information for leading the user to adopt a healthier lifestyle, like doing regular exercise or quitting smoking. Using video acquisition, this system detects users facing a mirror using two cameras, and uses face tracking algorithms to track faces in the collected images. After this process morphing techniques are applied to the captured face image to produce aging effects. This system provides information about future consequences of the user present behaviours. The results are tailored to each individual and the system uses image enhancement to provide each user with images of their aging process to persuade them to adopt healthier living styles.

Figure 2.2 – Behavioural effects in Persuasive Mirror project from [5], Figure 1

Choosing the right time for technology to make a suggestion is crucial. Studies show that just-in-time motivation messages are one of the keys to persuade people to change. Intille [6] stated that an effective strategy to motivate behaviour change is to present a simple message that is easy to understand, at just the right time, at just the right place, in a non-annoying way. This assumes the use of sensors and a way to provide different stimuli, to allow the selection of the most appropriate time to present relevant information given the context. Arroyo et al. [7] also propose the use of "just-in-time" prompts right at the point of behaviour in response to user's activities, with the use of simple and explicit messages in a non annoying way. Although it is not easy to know and choose the exact moment to present a suggestion, people tend to be more predisposed to change their attitudes when they are in a good mood, when they find that their current view no longer makes sense and when they feel in debt because of a favour they received [2].

Furthermore, computers allow the possibility to provide self-monitoring or surveillance to the users. There are several products and prototypes using self-monitoring strategies. Toscos et al. [8] developed ChickClique. This system's intent is to help teenage girls to change their exercise and eating habits. ChickClique helps girls to choose what to eat and provides them with the caloric values that they burn, by counting the number of steps they walk during the day. These calculi are compared to other girls' results in a group of friends using ChickClique. This comparison between values provides healthy competition between the girls which helps to obtain better results. Another example is UbiFitGarden [9]. Using on body sensors, UbiFitGarden controls physical activities such as cardio, walking, resistance training, or flexibility training. The system is able to schedule a weekly plan that can be consulted in a mobile device. While users perform their physical activities according to the plan, the system displays flowers blossoming in a garden until butterflies appear which signals that the program is complete. Self-monitoring applications are used mostly on health areas, monitoring user's habits and establishing goals for them to improve their health condition.

While self-monitoring allows users to know the consequences of their own actions, surveillance is used to monitor others' behaviours. People are more careful with their attitudes when they know they are being watched. This can cause changes in attitudes and/or behaviours. For instance, persuasive systems like CleanSink [7] use surveillance to persuade users to keep their hands clean. This system was designed for places that required employees with absolutely clean hands, like hospitals, restaurants or industrial clean rooms. CleanSink uses a camera to sense hands at the sink. The system is combined with an RFID card reader that logs the user's identity and compliance to a central database. When the camera detects that user's hands were under the sink long enough the RFID reader registers the compliance on the database.

Figure 2.3 – Clean Sink showing indicator (left), RFID reader (middle) and sink (right) from [7], Figure 3.

CleanSink was implemented in two different places, in a hospital and in an industrial clean room. In the hospital, the sink sensor was connected to the lights and the users had to wash their hands or they would not see the room clearly. In the industrial room the sensor was connected to the door and users had to wash their hands or they would not be able to leave the room.

The conditioning theory is based on the fact that humans are ambitious. Setting goals for the interaction and rewarding users with positive reinforcements motivates the users to perform well. Niebuhr et al. [10] presented a study that explores this kind of motivation technique. When users need to perform repetitive tasks it is usual that they get bored while performing it. When using an interface that displayed goals to be achieved on a simple task, like inserting elements in a database with the use of a form, users performed their task without distractions. This study reveals that even with the use of simple messages, such as displaying the number of tasks left to complete the work, users show a higher motivation to complete their work when they are aware of the goals previously set and their work progress.

2.2.2. Social Awareness and Support

When users interact with computer systems that appear to provide intelligent feedback to the actions they perform, computers resemble human characters and can be seen as social actors. Computers can have a social influence and people can relate to them as they would with a human being [2]. As social actors, it is important that when people are interacting with a computer system it reinforces compliance with positive feedback to keep people open to the new experience they are facing. There are many characteristics that computers may possess which can lead people to treat them as social actors as shown in Table 2.1.

Table 2.1 – Computers as Social Actors characteristics, based on [2], Table 5.1. , page 91

Characteristics	Examples
Physical	Face, eyes, body, movement
Psychological	Preferences, personality, emotions
Language	Interactive language use, spoken language, language recognition
Social Dynamics	Cooperation, praise for good work, answering questions
Social Roles	Doctor, teammate, opponent, teacher, pet, guide

Persuasive system's behaviours and interfaces can get closer to humans', Baby Think It Over [11] has the form of a human baby and simulates babies behaviours to persuade teenagers not to get pregnant. Also, to increase social influence, it is reasonable to suggest that an attractive interface will have greater social influence than an unattractive one. Attractiveness can, therefore, be faced as a persuasive element [2].

Sometimes, computer software seems to have personality, emotions or preferences. The way messages are shown can make computers look polite or even angry. They can be programmed to react to some kind of interaction giving the impression that they look alive as they respond differently according to each situation.

Computer software can be very flexible, which allows having multiple language forms related to target users. With the use of the appropriate language, it is possible to reach each particular user, making it easier to persuade them. Likewise, it is a good politic to praise users when they do what they are supposed to do and help them through cooperation. "By offering praise, via words, images, symbols or sounds, computing technology can lead users to be more open to persuasion" [2]. Also, mixing praise with few non-praise messages increases the credibility of the first ones.

Applications can also assume social roles, being seen as having professions or character like persons do. Although it is not a persuasive system, "Tamagotchi" successfully demonstrated how interacting with a computer could be a social experience. For instance, McCalley et al. [12] developed a system that was directed to the elderly to create an emotional bound between them and a pet plant. The pot in which the plant was inserted responded to human stimulus. The Pet Plant system covered four experiment conditions:

- Control Plant: the system blinked in a random colour and intensity and did not respond to human presence or action.
- Light Plant: the system reacted to human touch and voice, changing its green colour intensity while users held the plant pot and turn into red and green simultaneously, creating a "blush" effect, if the users touched the plant's leaves.
- Similar Plant and Dissimilar Plant: the system reacted the same way as the Light Plant with the difference that the first had a card describing it as being shy and the second had a card describing it as being extroverted.

The group of testers was randomly assigned to one of the four experiment conditions. The users were told to read the card before the interaction. Results revealed that the users attributed emotion to the plant, and more significantly in the last three experiment conditions, with Similar Plant having the best results. Also, Control Plant had the worst results when users were asked if they liked the plant. The other three experiment conditions had similar results, with Light Plant having slightly better results than the other two. This study reveals that when using the right interface the system can be seen has being alive and therefore having intelligence and feelings.

Simultaneously to the fact that computers can be seen as social actors, persuasive systems can develop social conscience on users. Reitberger et al. [13] developed a persuasive system, PerCues, intended to reduce pollution through the use of public transportation. If people choose to use the bus instead of the car, the system informs the user, through a mobile phone, about the consequences of their own choice. The information about all the users that individually had the same decision is gathered and the group of persons who had the same choice know from that moment on the impacts of individual and group action. Using individual actions to inform about group state can be a good form of persuasion because people relate to a more significant change with a simple action.

2.2.3 Credibility in Persuasive Systems

"Simply put, "credibility" can be defined as believability. In fact, some languages use the same word for these two English terms. The word credible comes from the Latin credere, to believe." [2]. Persuasive systems have to be credible, only this way people will rely on the information which has the intent to persuade them. Also, Fogg defines credibility as having two dimensions: Trustworthiness and Expertise.

Trustworthiness is, in context of computers, a computer that is perceived to be truthful, fair and unbiased. One the other hand, *"Expertise consists of those characteristics, skills and knowledge of a person (that is, expert) or of a system, which distinguish experts from novices and less experienced people."*[2] These two dimensions do not necessarily have to come together, but combining them leads to more credible sources. In computer systems the

[2] Hhttp://en.wikipedia.org/wiki/Expertise

information can also be provided by credible third-party sources which will add credibility to the computing technology.

Fogg [2] also states that even though there are some cases where the credibility does not matter, there are others where credibility is essential in human-computer interaction. Credibility matters when computers:

- Instruct or advise users.
- Report measurements.
- Provide information and analysis.
- Report on work performed.
- Report about their own state.
- Run simulations.
- Render virtual environments.

To conclude this section, it is important to refer that the presented design principles for persuasive technology, that Fogg conceptualized, described in section 2.2.1, 2.2.2 and 2.2.3, have to be analysed as software requirements [14].

2.3. Ubiquitous Systems

"*The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.*" [15]. This vision is being considered now more often than ever, technology advancing into the future being used on everyday things.

Since computers came into people's lives, each one had to learn how to work with them, finding it often frustrating. Mark Weiser brought a new vision to Human-Machine Interaction residing on the idea that instead of one having to learn how to work with computers, it should be the opposite, computers should merge into human's lives being as much invisible as possible. This concept was named "Ubiquitous Computing" and it has two main objectives:

- Detect and interact with people in the system's surroundings.
- Blend in the environment.

The concept of ubiquity is refreshing. People tend to find computers frustrating and intrusive, however if systems detect human presence and allow people to act like they normally do, the interaction arises naturally.

2.3.1 User Aware Environments

When developing persuasive environments, it is indispensable to analyse users' behaviours to track changes and to evaluate if they are really doing what is expected they would do. Only this way it will be possible to conclude the product's success. Nevertheless, it is essential to acquire information of people's behaviour to determine their needs and build a system based on those. *"Differences between expectations of how people will behave and how they actually do behave in the complexity of real settings contribute to product failures. Developers often make erroneous assumptions about the need for, use of, and reaction to new technologies."* [16]. Besides this, persuasion in ubiquitous environments can only be present if there is a way to analyse people's behaviours. It is possible to do so using three different kinds of approaches:

- Gathering information through direct observation.
- Having users submitting information.
- Using the system's sensors to gather information.

Direct observation, can be very time consuming and by that fact it is most of the times considered the last approach to use.

Having users submitting information about their behaviours can be done, on regular or random time intervals and on system's response to user initiative. To submit the information, one could have, for example, users responding to a questionnaire provided by the system. While gathering information on regular or random time intervals requires a timer, having the system responding to user initiative requires sensors to detect the right time to present the questionnaire. As for all these methods the questionnaire could interfere with the interaction, in case the system's designer uses one of these kinds of approaches he has to put a lot of effort into the interaction model to keep users satisfied.

The most complex approach, but also the most effective one, as it does not distract the user from his main goal, is to use sensors that respond to user initiative and that can gather information about their behaviours. The sensors used, where they are placed and the information to collect has to be chosen taking into account the behaviours to be changed and the environment in which the sensors will be applied.

Chi et al. [17] developed a Smart Kitchen with the intent to follow this last approach. This system's purpose is to enable calorie awareness for families to have healthier meals. Smart Kitchen was developed with hybrid sensing to calculate the number of calories in the prepared food. To do so, the system has a weighing-sensing surface deployed on the kitchen counter and on the stove, and a camera to capture images to detect the food. Although it has some limitations, it has 92% recognition accuracy of the prepared food caloric values. The recommended calories for the meal and the calories used in each moment are displayed in front of the cookers, so they always have a perception of the current state and have a glance whenever they want. The tests for this prototype involved three participants who could choose the menu to cook. They prepared their meals with and without the system's help. In all the cases the food preparation without the help of the system had higher values than the recommended. With the system's help participants approached the recommended values for the number of calories on each meal. All the participants wanted to change their cooking behaviour and therefore the results were pretty satisfactory. This work's implementation is based on an architecture presented on Figure 2.4 bellow.

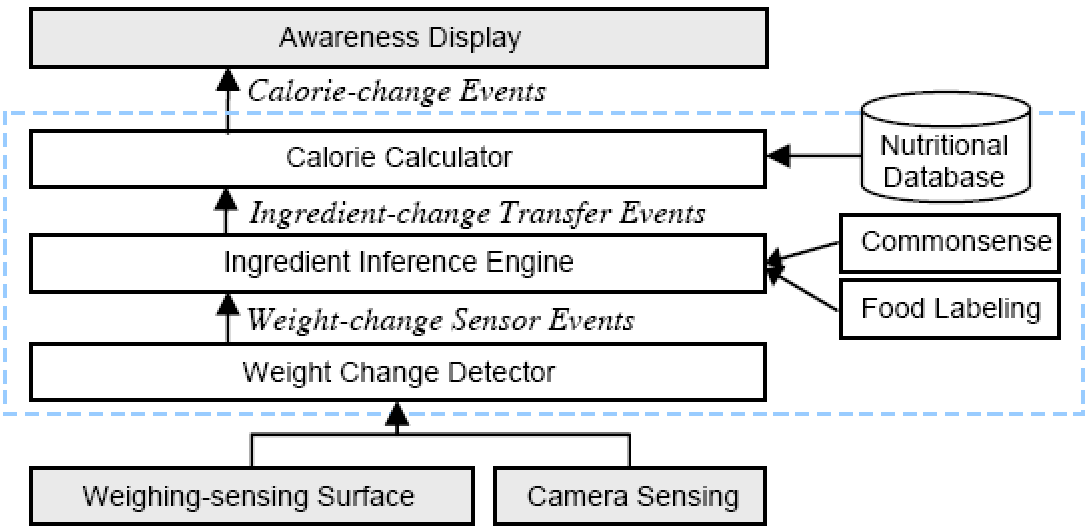

Figure 2.4 – Calorie tracker architecture from [17], Figure 2

This architecture is based on hybrid sensing as it has a weighing-sensing surface and camera to detect the environment's context. Ingredients' weight is gathered by the weight-sensing surface, and the type of food present on the cooking area is provided with the use of images collected by the camera. After being determined the weight and the types of ingredients currently used, either by the Ingredient Inference Agent analysis on weight and food transfer or by an observer input that catalogues the food in use, the data is sent to Calorie Calculator. The calories present on the food in preparation are then determined and the results are shown to the user who is preparing the food, so that he has an idea of the caloric value when preparing a meal.

The system presented by Timothy Bickmore et al. [18] aims at helping people to obtain a healthier life style. This PDA-based health advisor provides real-time reminders and conversational counselling to help users to change their health behaviour. The architecture developed for this project (Figure 2.5) is the base to support real-time messages according to time and sensor data that manages user's behaviour.

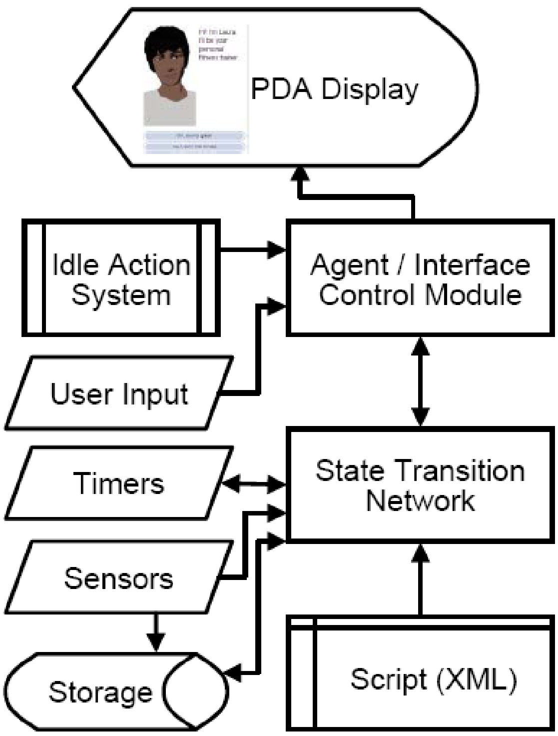

Figure 2.5 – Health Advisor runtime Architecture from **[18]**, Figure 2

This system displays, on the PDA screen, an animated agent that simulates emotions and "intelligent behaviours" like facial expressions, head nods, eyebrow's movement, posture shifts, eye gaze movements and a nonverbal conversational behaviour. User inputs are confined to multiple choice selections and time-of-day specifications. According to these settings, interaction dialogues are scripted in a XML-based state-transition network. The scripts consist on agent utterances, user allowed responses to each agent utterance, instructions to state transitions based on these responses and other system events like timers, and sensor inputs. The actions are controlled by a finite-state machine built at runtime and the messages are displayed consistently to what is defined in the XML script. During the time in which the script does not require agent actions, the system becomes idle and displays idle behaviours like eye blinks and posture shifts.

2.3.2. Environment's Context Detection

The way to interact with users in ubiquitous systems is a big design concern. As people do not like intrusive systems, it is necessary to develop an interaction model that subtly increases users' participation. Consequently, it is important to divide interaction in different phases and adequate messaging type for the context in question. Vogel et al. [19] proposed an interaction method for ambient that goes from implicit to explicit interaction and from public to personal displays, motivating users to get close enough to interact directly with system's information.

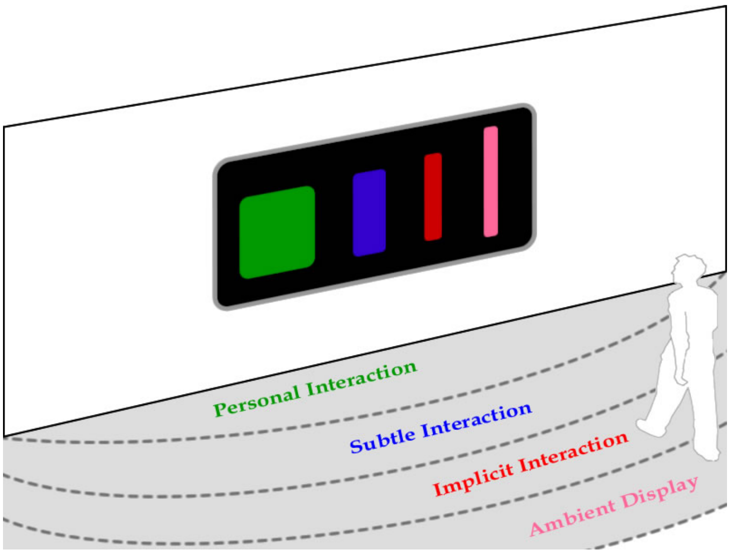

Figure 2.6 – Public to Personal Interaction Model based on [17], Figure 2

The framework presented consists in four interaction phases, Ambient Display, Implicit Interaction, Subtle interaction, and Personal Interaction, being the order given the same to the system display proximity as shown above in figure 2.6.

In phase one, Ambient Display, an overview of the system is displayed using appealing information, being easy to identify the context of the system, without giving to much information. In the second phase, Implicit Interaction, the system should recognize that users are facing the screen, showing an abstract drawing of their faces. Also, personal or public information which is important for that moment is subtly shown since these techniques help to get users closer to the display. The next interaction zone is Subtle Interaction. When a user pauses for a moment in this interaction zone, features are displayed in more detail. Here users may select the information to investigate with simple hand gestures while other users can continue to see “their” part of the system. Since users are closer to the system, it can display personal information. This information has to be harmless, since it can be seen by others compromising user’s privacy. After selecting the information to manipulate, users reach the last phase, Personal Interaction. Here, information can be directly manipulated touching the screen and as the user’s body is in front of other users view, it can help to occlude personal information that could be presented. Nevertheless, all other information must be visible to users interacting in other phases giving the possibility to multiple simultaneous interactions. Furthermore, transitions between states are very important. These transitions must be discrete so users do not notice an abrupt change from one phase to the other, and at the same time, not too subtle that users do not realize that they reached a new state. In the first two phases, the system’s states change according to users’ movements like approaching the nearer interaction phases or facing the sensors. User’s actions become more explicit in the last two phases, choosing information to access with hand gestures in the Subtle Interaction phase and touching the screen to directly alter information in the Personal Interaction zone.

Another interaction method has been proposed by Streitz et al. [20]. “Gossip Wall” was designed for an office environment having particular information needs both at the collective level of the organization, and at the personal level of the worker. The interaction zone was divided in three phases: Ambient Zone, Notification Zone and Interaction Zone.

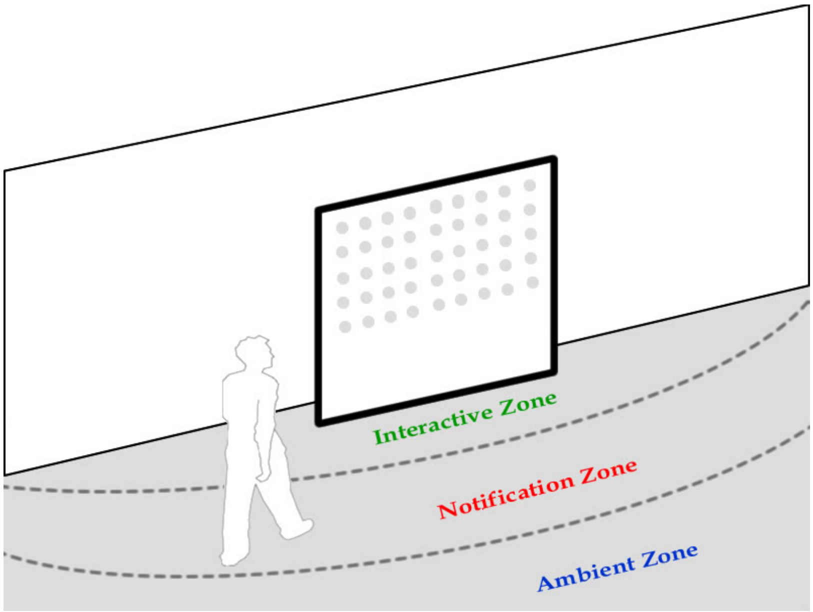

Figure 2.7 – "Gossip Wall" Interaction Model based on [20], Figure 1

While people are outside the system's sensor range, Ambient Zone is displaying its message. The Ambient Zone messages are individual independent, which means that they are the same for all the users. When users reach the Notification Zone, Gossip Wall changes into a notification status and displays individual or group information in view of persons in system's range. Users can browse, download or enter information, for further use, using a mobile device. While they get closer to the wall users reach Interactive Zone and they can interact directly with independent interactive pixels that store information sent throughout the mobile devices.

These phased interaction methods allow users to become more participative as they comply with system´s goals without feeling intrusiveness from it.

2.3.3. Persuasion in Ubiquitous Systems

One approach to follow in persuasive systems is described by Oinas-Kukkonen et al. [14]. The paper in question addresses the fundamental issues behind persuasive systems and the way to evaluate them. It defines seven ideals that need to be taken in consideration when developing a persuasive system:

- *Information technology is never neutral.* Information always influences people and so persuasion should be, if possible, a process instead of a single act.
- *People like their views about the world to be organized and consistent.* In short, this relates to the fact that if users feel any kind of inconsistency in their reasoning they will reflect on their current views and probably will adopt an attitude change.
- *Direct and Indirect routes are key persuasion strategies.* In this information era it is difficult for people to process all the information they receive. This implies that only users that are really interested in the information provided by the system will pay their full attention and process all the information. For the others, the system could trigger memories represented by images, words or some kind of information that the system displays, assuming that the new information has an influence on user's background.
- *Persuasion is often incremental.* This means that in certain situations, as it is not easy for a person to change habits from one day to the other, the tasks can be slowly increased, so its effect duration lasts for a longer period of time, permanently if possible.
- *Persuasion through persuasive system should always be open.* The information should always be truthful so it does not end-up misleading the user.
- *Persuasive systems should aim at unobtrusiveness.* User's attention should be focused on their main tasks. Everything that comes in between can divert user's concentration which will affect the persuasion effect.
- *Persuasive systems should aim at being both useful and easy to use.* If the system is hard to use, it can frustrate users and will result in a very negative aspect for persuasion. On the other hand, if the system is easy to use it could be used by almost everyone without much effort and it will help the persuasion process.

As described above, a property desired for ubiquitous systems is to blend in the environment. There is a big tendency to develop persuasive systems with this principle, mainly because persuasive systems are normally designed to interact directly with persons but they should be seamlessly integrated in the environment, not significantly changing it.

2.4 Human-Computer Interaction

"Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them." [21]. The obvious intervenient on this field are Humans and Computers, so one must [22]:

- Understand *computers*:
  - limitations, capacities, tools, platforms

- Understand *people*:
  - psychological, social aspects, human error

As it is not linear to design interactive systems, one has to know what Human Computer Interaction (HCI) aims at and the principles to achieve these goals.

2.4.1 Human-Computer Interaction Goals

A basic goal of HCI is to improve the interaction between users and computers by making computers easier to use and receptive to the user's needs. Jennifer Preece et al [23] identify two main types of goals for interactive design: Usability goals and User experience goals.

Usability goals are centered on the system’s use. Interactive system’s goals are:

- Effectiveness – how good the system is doing what it is supposed to do.
- Efficiency – the way system supports users in carrying out their tasks.
- Safety – user protection on dangerous conditions and undesirable situations.
- Utility – when systems provide the appropriate functionalities so users may perform their tasks.
- Learnability – how easily users learn to use the system.
- Memorability – how easy the system is to remember how to be used once learned.

User experience goals are centered on users' experience when using the system. The system should be:

- Satisfying
- Enjoyable
- Fun
- Entertaining
- Helpful
- Motivating
- Aesthetically pleasing
- Supportive of creativity
- Rewarding
- Emotionally fulfilling

These goals are the guidelines to develop efficient interactive systems to minimize user's difficulties and barriers on interaction.

2.4.1 Interaction Design Principles

When designing a new interactive system, there are many aspects to have in consideration. It is important to have in mind the following design principles [22]:

- **Early user focus**:

  The users to whom the system is going to be developed for have to be studied in advance. When designing new interfaces it is indispensible to know who the users are, they probably don't have the same interests and the abilities of the developers and it is important to watch them in action and determine their needs.

- **Empirical measurement**:

  One should watch the users in action. It is important to establish quantitative usability specifics such as the number of users performing the tasks, the time to complete the tasks, and the number of errors made during the tasks.

- **Iterative design**:

  After determining the users, tasks and requirements, the system design should follow a cyclic process allowing for the early detection of errors and their respective correction at reduced costs.

The iterative design process should be repeated until a sensible, user-friendly interface is created. This is a key process to develop interfaces. This cycle is represented below in Figure 2.8.

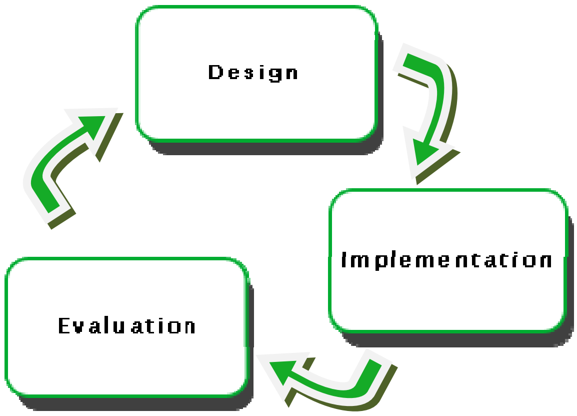

Figure 2.8 - Design Cycle based on [22]

Iterative design is the most used process for developing user interfaces. It is a specification of the spiral model (Figure 2.9) described by Boehm [24].

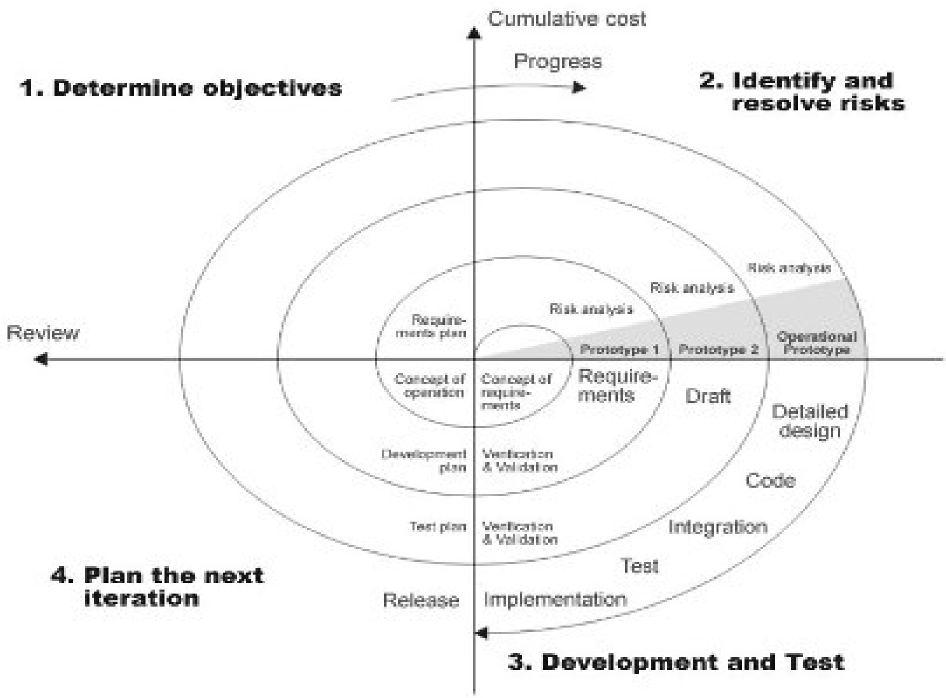

Figure 2.9 – Boehm’s spiral model from [24]

The designed software is improved at each cycle and following this spiral model, at each iteration one approach the operational prototype. The cost of each iteration step is directly proportional to the radial dimension of the spiral model.

2.5 Child-Computer Interaction

One intuitively may say that developing systems and applications for children is different than for adults as it has to do with child psychology and learning capabilities. Janet Read et al [25] define Child Computer Interaction has follows. "*The study of Child Computer Interaction is a growing subfield of HCI. Child Computer Interaction encompasses traditional HCI but also specifically reaches out into the areas of child psychology, learning and play*." Hence, one has to understand what motivates children, to develop new systems according to their needs. Evaluation techniques should also be adapted to children specific characteristics. Janet Read has studied different techniques to measure children's opinion of technology [26].

2.5.1 Motivation and Goals

Understanding what drives children to compel with their tasks is crucial when creating new interfaces. Motivation takes a decisive part in children's learning process. Children lack the motivation for long and medium term objectives. Thus, short term objectives are essential when creating software directed to them. For this reason, immediate feedbacks are indispensible to motivate children to compel with their tasks. Instead of traditional learning process, like in school classes, which has no immediate rewards, a game creates the possibility of positive feedbacks when children accomplish their objectives. Bearing this in mind, each decision made in an application must have a well-defined inherent goal and respective feedback in order to increase their motivation and will to complete the tasks.

To produce enjoyable interfaces one has to first think about what users want. In the mid-1970s, in an attempt to explain happiness, Mihaly Csikszentmihalyi introduced the concept of Flow. “*Flow represents the feeling of complete and energized focus in an activity, with a high level of enjoyment and fulfillment*” [27], which means that once experiencing this feeling, one is fully motivated and focused in what is being done. Csikszentmihalyi’s research and personal observations identified eight major components on Flow [28]:

- A challenging activity requiring skill;
- A merging of action and awareness;
- Clear goals;
- Direct, immediate feedback;
- Concentration on the task at hand;
- A sense of control;
- A loss of self-consciousness;
- An altered sense of time.

Although there are eight major components on the Flow, it is not necessary to comply with all of them. Nevertheless, the experience should be immersive and cover as many of these components as possible. Reaching and maintaining the Flow is a big challenge. The graphics 2.10, 2.11 and 2.12 show how one may keep users in the Flow zone [28].

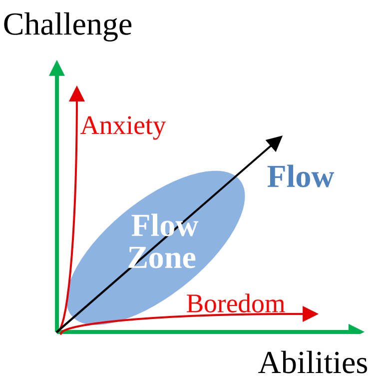

Figure 2.10 - Flow Zone factors.

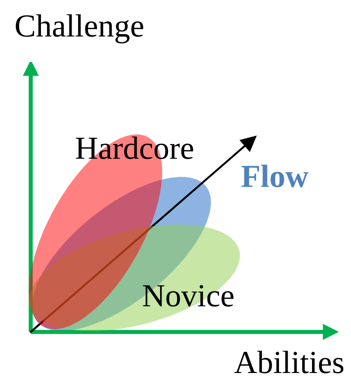

Figure 2.11 - Different players have different Flow Zones.

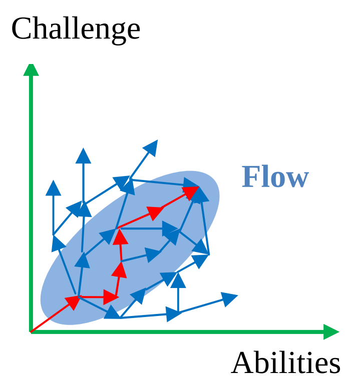

Figure 2.12 - Designers adapt players’ flow experience through the choices they deliberately build into the experience.

The idea is to balance the challenge provided to the users with their abilities. Hence, one should:

1. Mix and match the components of Flow;
2. Keep the user's experience within the user's Flow Zone;
3. Offer adaptive choices, allowing different users to enjoy the Flow in their own way;
4. Embed choices inside the core activities to ensure the Flow is never interrupted.

3. Methodology and framework description

This work focuses on the development of a framework to support the development of contextual aware persuasive applications. The achieved solution came as a result of the study presented in chapter 2. One found the need to develop an architecture that followed the line of persuasive technology. Hence, the architecture had to provide feedback to user's actions in order to motivate their behaviour's change. Furthermore, as the proposed solution aims at allowing the creation of new persuasive environments, that call for ubiquity, the architecture was created in such a way that the sensors are indispensable tools for user interaction. The architecture also allows for multimedia data output that can be configured according to the users' actions.

In the scope of Human Computer Interaction, the necessity of providing a simple way to produce functional interfaces lead to the creation of an authoring tool of simple use, which grants to its users the possibility of having an effortless way to develop their applications. Thus, the developers can centre their attention more on the design and in the evaluation of the desired final result and less on the implementation details. Hence, an authoring tool was developed to support this feature. Although the main purpose of the authoring tool was to facilitate the creation of new applications, the prototype implementation is not directed to all users and one will further present the evaluation of this tool's usability and the potential of this framework.

As a proof of concept, one designed an application, Smart Bins, which was developed to promote recycling. The purpose in the creation of this game environment was to teach children about this subject, motivate them to adopt a different conscience on this matter and encourage them to recycle. The idea to develop a game came with the intention of motivating children in short-term periods. In addition, one would also provide them with an amusing and easy way to learn more about recycling and therefore change their attitudes and behaviour about this theme. Moreover, this application was studied in terms of usability and to better understand the potential persuasiveness effect of one such application created using this framework.

3.1 Framework Design Methodology

The framework developed in the scope of this master thesis aims at allowing the creation of ubiquitous persuasive applications with a minimal programming effort, so non-programmers can build context-sensitive persuasive applications. These applications are designed to be able to capture contextual information regarding the user's behaviour or the surrounding environment and react accordingly, in order to persuade users to adopt a certain attitude or behaviour. In section 4.2, it is described one such application developed with this framework. However, this framework can be used to build a great variety of different ubiquitous persuasive applications, such as advertising installations or augmented physical objects. To design the framework the concepts of *captology* [4] and *ubiquitous computing* were combined, since the goal is to facilitate the authoring of non-intrusive persuasive applications that subtly guide the users through planned appropriate actions that change their attitudes or behaviours. The work of Oinas- Kukkonen and Harjumaa [14], which is based on the work of Fogg [2], was also taken in consideration. It defines persuasive design strategies and guidelines, listing techniques for persuasive system content and functionality.

The proposed architecture seeks to contribute to the study of how computer systems can be used as persuasive technologies to change people's attitudes and behaviours. Sometimes people do not understand what they have done wrong and they need to be warned about their inappropriate behaviour, in order to be able to change their current ideals and attitudes, leading to the adoption of a different, more suitable behaviour. Nonetheless, a persuasive system has to be both attractive and non-intrusive, capturing the characteristics of the surrounding environment, detecting the user's attitudes and distinctiveness in order to respond accordingly. This chapter describes the framework design, explaining the conceptual solution for this work. The proposed architecture is supported by sensors to detect users and their context in the system vicinity. The set of characteristics collected within the system vicinity is translated to different actions through the usage of an interaction model. Consequently, for distinct patterns detected by sensors, different multimedia resources can be provided to users, also making the interaction model to change state according to the sensors' pattern detected.

This framework consists in two distinct parts: an authoring tool to develop new persuasive applications for ubiquitous environments and an engine that builds these environments and manages user interaction according to what was defined when using the authoring tool to create these applications. Figure 3.1 shows the development process to create different persuasive applications using this framework. This process may be repeated as many times as users want and matured until the developers of the new applications are fulfilled with the result, since this authoring tool permits to easily edit the already created configuration files.

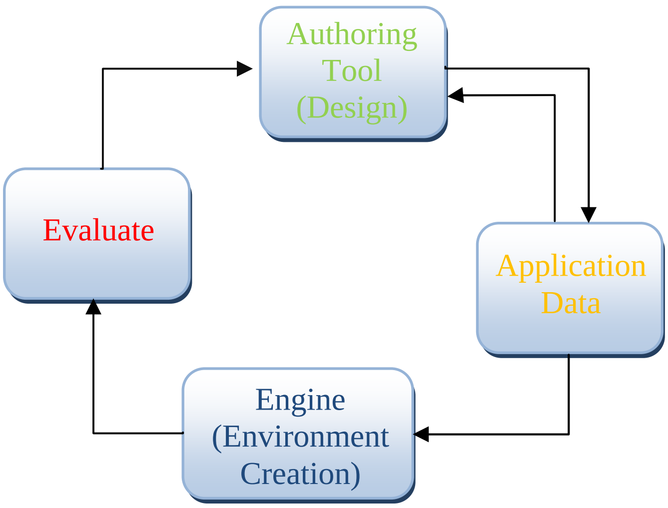

Figure 3.1 - Application Development process

In order to explain the design of the proposed solution, this chapter is structured as follows. First, the framework architecture will be presented in section 3.1.1. Section 3.1.2 is reserved to explain how to design new applications with the use of the authoring tool. At last, in section 3.1.3, a diagram with the system's classes will be presented and discussed.

3.1.1 Framework Engine Design

The proposed architecture relies on four main modules: Multimedia Display, Context Processor, Data and Sensors. The diagram shown in Figure 3.2 represents the different modules and shows how they exchange information between them.

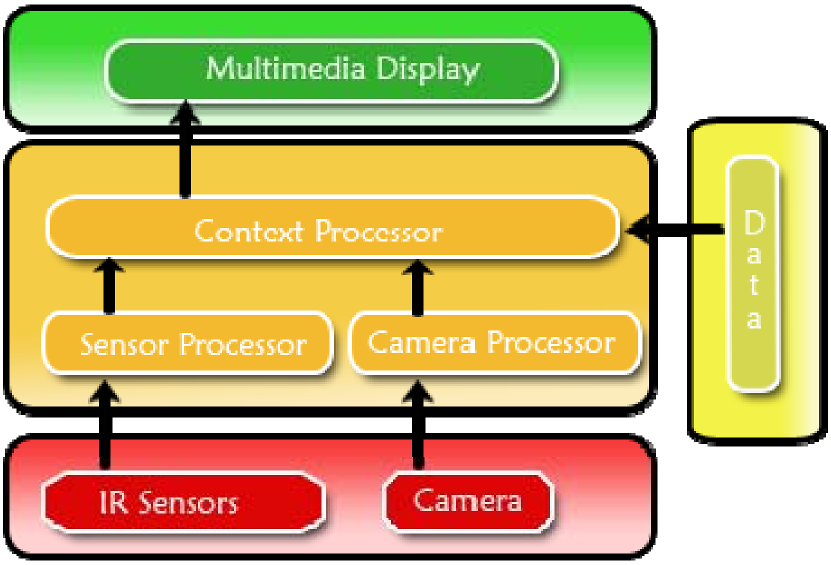

Figure 3.2 - Framework Engine Architecture

This architecture allows the system to evolve through time given that the modules are independent, relying on common interfaces to exchange information between them. Therefore, new features can easily be integrated within each module to fit the needs of each application. The Context processor is the main module of this architecture as it has to manage all the information provided by the other framework modules. When the sensor processor and the camera processor receive the information from the sensors and camera respectively, both modules filter the data and send it to the Context Processor. Then, the Context Processor analyzes the data provided by the lower layers, accesses the application data to gather additional information and presents it through the Multimedia Display. Finally, the Multimedia Display is the module that provides the user with the result of each interaction with the system. Visual and audio stimuli are the primary output of the system. Output stimuli result from the analysis of contextual information, gathered by the sensors and the camera. This analysis is performed within the context processor and, afterwards, that stimuli are shown to users within the multimedia display.

3.1.2 Framework Application Design

The engine described above is supported by an authoring tool that eases the development of new persuasive applications. The authoring tool explores the persuasiveness achieved by this framework and facilitates its usage in different contexts. During the usage of the authoring tool, files are configured so that engine's modules may interpret the information. This information is added to the platform in order to be used within the new application, building the interaction model for that application. Moreover, existing interaction models can be further modified and adjusted to specific purposes. Therefore, while the framework can evolve over time, new applications can also evolve to integrate new features added to the framework. The authoring tool explores three major concepts: frames, resources and links. Figure 3.3 describes how the diverse framework elements evolve during the execution of an application. Each column shows the correspondence between the elements and how they evolve in order to persuade a user to adopt a new behaviour.

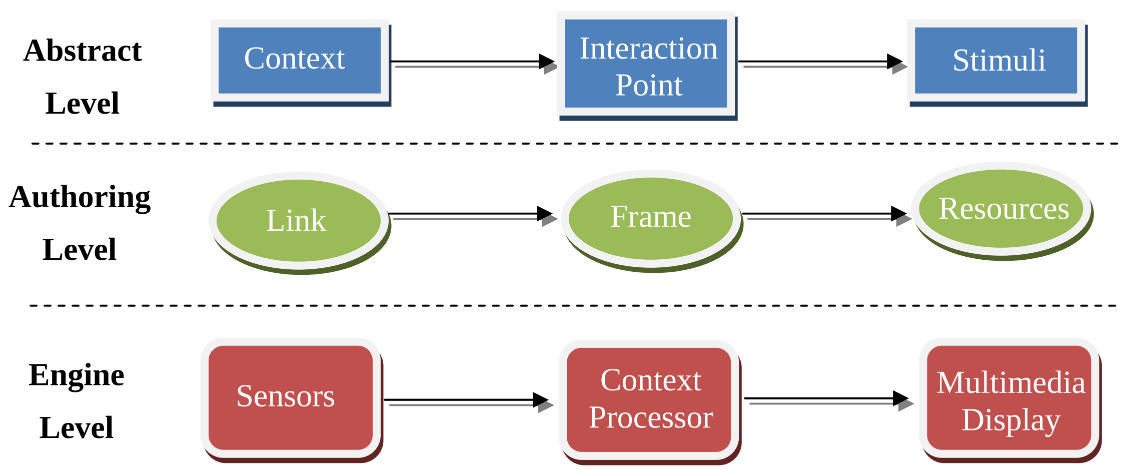

Figure 3.3 – Elements evolution in the framework.

Frames correspond to the major interaction point in an application, since they reflect an application state within the interaction model. A frame aggregates the remaining two concepts: resources and links. Indeed, resources correspond to the media used to compose the frames and are able to provide the stimuli to the user within the application. Resources can be images, videos, texts, audios, animations or other media able to persuade the users. Links correspond to application state transitions. A link establishes a transition from one frame to another and defines the conditions (set of context variables) that activate that transition. When a specific sensor pattern, matching the condition to activate a link in the current application frame is achieved, the corresponding link is activated and the application automatically changes to the interaction frame defined by that link. While defining a frame, one needs to specify its media content and define the possible links to other frames.

In summary, the application design can be seen as an oriented-graph, where the nodes correspond to the frames, which contain the multimedia resources and the edges correspond to the links that can be followed from one frame to another. This way, Figure 3.4 represents an example of what could be an application design model using this framework.

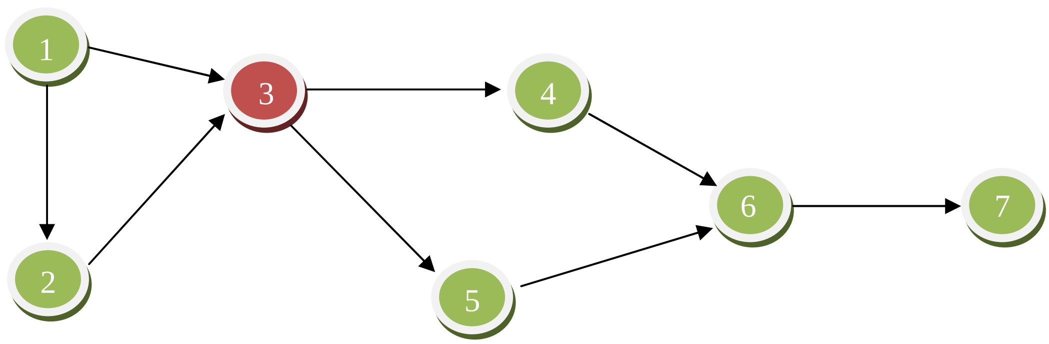

Figure 3.4 – Design Model for an Application

If one thinks of this graph as the application model design at runtime, the red node represents the frame that corresponds to the current application state, and the green nodes represent the remaining possible frames of the application.

This framework is generic and allows the creation of multiple applications, but it requires that the user who is responsible for creating an application has a well formed idea of how the application will function. Moreover, all the multimedia contents for the frames have to be available in advance. Otherwise, the authoring process must be interrupted in order to collect the required data.

3.1.3 Framework Class Diagram

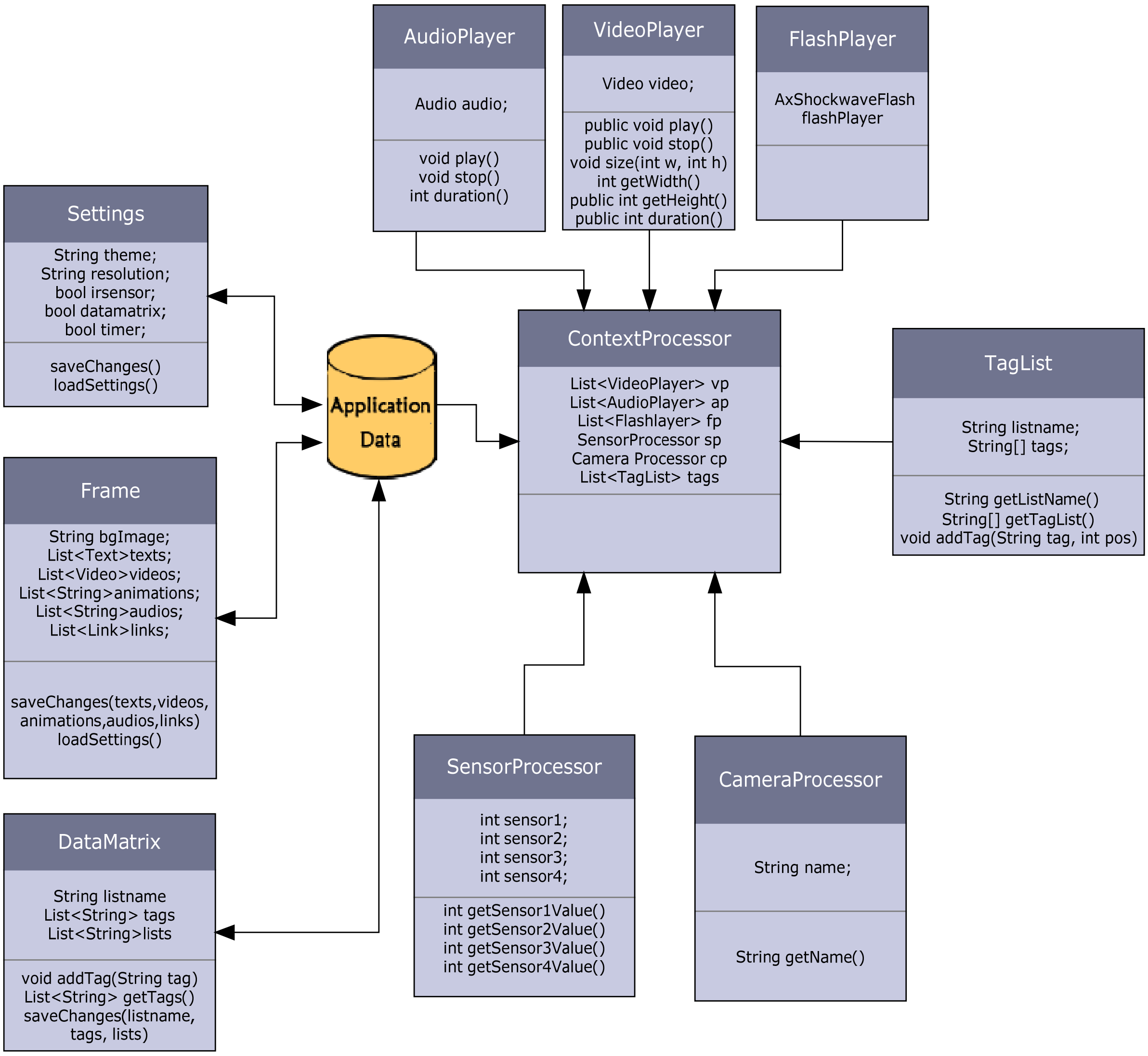

Figure 3.5 – Framework's Class Diagram

This class diagram represents the most important classes and methods that suit the framework's best interests. The classes presented on the left side of the Database - Settings, Frame and DataMatrix – are the authoring tool's components. On the right side of the

Database, the classes represent the most important classes of the framework's engine. Now we describe, in more detail, the different components of the diagram:

Authoring Tool Classes:

Settings: this class stores the application's general settings – theme, resolution, infrared sensor use, data matrix use and timer use.

Frame: as described earlier, a frame is the concept that aggregates the multimedia resources, such as videos, flash animations and audio to be present in a specific application state. Each frame also holds the different links to other frames.

DataMatrix: the tags to use in each application are stored through the use of this class.

Engine's Classes:

ContextProcessor – the context processor manages all the engine's resources. As the central and crucial part the framework, it has to manage all the frames, resources and links in order to evaluate the contextual situation and present the correct stimuli to the users.

SensorProcessor – this module manages the data provided by the infrared sensors. The data is then processed and converted to the distance to the sensors, in centimetres.

CameraProcessor – the frame images are collected with the use of a camera and the images are also analysed to detect data matrix codes.

TagList – this class supports the context processor use of the data matrix codes. At the start of each application the context processor loads the tags associated with the application and stores them in the TagList class.

AudioPlayer – this class manages the audio resources used to provide the necessary stimuli for the application's users.

VideoPlayer – this class loads and controls the video resources used to provide the necessary stimuli for the application's users.

FlashPlayer – this class manages the presentation of flash animations.

4. Framework Implementation and prototype development

In this chapter the implementation is discussed in more detail. One focus the implementation description of the framework components and the approach selected as a proof of concept for this framework.

The framework implementation consists in two main components: the engine and the authoring tool. The engine implementation aims at providing a generic solution to create and manage persuasive environments, guiding the user through an interactive experience. To create the necessary configuration files for the engine to build the persuasive applications, one implemented an authoring tool. Furthermore, the implementation of the application developed as a proof of concept for this master thesis, Smart Bins, is also described in this chapter.

4.1 Framework Implementation

One of the main focus and contributions of this master thesis is the developed engine. In order to support it, an authoring tool was also produced to provide a simple way to create the configuration files which will result in the applications. The authoring tool, whose implementation will be described in detail in this chapter, was developed using C#. It uses forms to guide the users in the creation of the new applications. According to the definitions introduced through the form menus, the necessary configuration files to be used by the engine are generated. This allows the creation of the context aware environments. Besides the Application's General Settings menu, Frame's menus must be filled with the resources and links configuration for each possible state of the application. The current engine implementation was also developed in C# programming language and will be described in the following section.

4.1.1 Engine Implementation

The engine was implemented so that the XML files created by the authoring tool could be read in order to create the desired persuasive environment's applications. The engine works in a way that each module is independent from the others. In order to better understand how the framework was implemented, this chapter explains the most important modules of the engine, Sensor Processor, Camera Processor and Context Processor that together, with the modules that manage the multimedia resources bring life to innovative ubiquitous environments.

In general this engine works as follows. The settings for each application are loaded by the Context Processor module. The chosen resources are loaded by each responsible module for its use. AudioPlayer, VideoPlayer, and FlashPlayer modules for instance, manage audio, video and flash animations respectively and are controlled by the Context Processor. Context Processor also manages user interaction. As this module manages the data being sent by the lower layers modules, when it detects that the conditions of a link have been achieved it loads the next frame. Therefore, Context Processor waits for user interaction according to what was defined in the links. Hence, the context is always in line with what was defined as the possible changes for the application's frames. In the current implemention, four types of links can be defined to control the transition from one frame to another:

- **Infrared (IR) sensors –** These sensors measure distance. Thus, a certain distance threshold from a user or object to these sensors may be used to activate a link.

- **Camera images –** In the current implementation, the camera is prepared to detect Data Matrix patterns that can be used to identify items from the real world. A link can be activated when the camera detects a certain code. A **Data Matrix** code is a two-dimensional matrix barcode consisting of black and white "cells" or modules arranged in either a square or rectangular pattern. The information to be encoded can be text or raw data. Usual data size is from a few bytes up to 2 kilobytes. The length of the encoded data depends on the symbol dimension used. Error correction codes are added to increase symbol strength: even if they are partially damaged, they can still be read. A Data Matrix symbol can store up to 2,335 alphanumeric characters[3].

- **Time based –** It is possible to define a period of time after which a certain link is activated. In such cases, a new specific frame is shown after the defined period of time has expired. This allows the definition of links to be activated when no action from the user is required to change to another frame.

- **Number of visits to a specific frame –** This option allows the user to define to which frame the system will go after being in a specific frame a certain number of times.

4.1.1.1 Sensor Processor

Sensor Processor is the module that manages the data collected by the sensors. In the current implementation only infrared sensors are considered. When the links from one frame require the functioning of the sensors, they are activated to return the values from the distance to a user or an object. This framework is currently working with four infrared sensors that may be reused with a different purpose in each frame. The Sensor Processor emulates a serial port through which the data is sent. This data, received from the sensors, is filtered and converted to distance values, in centimetres, so that when it is required, the Context Processor can call the methods that return these distance values and use it to evaluate the contextual situation according to what was previously defined by the application developer.

4.1.1.2 Camera Processor

This module manages the data provided by the camera, which may also be considered a (special kind of) sensor. Whenever it is required, the Camera Processor captures an image

[3] Hhttp://en.wikipedia.org/wiki/Data_Matrix

from the camera and the data is converted to a Bitmap. DirectShow[4] library was used in the current implementation to capture the camera frames. Thus, it is possible to use a library that identifies the Data Matrix content from a Bitmap. The first library to be used to identify the data matrix codes was a library from the DTK[5] software for barcodes. As with this library the image processing tended to slow the application's flow, one tried another library for this purpose. One tested Barcode Lib[6], and this time the results were very satisfying so this library was used in the current implementation to detect Data Matrix codes present in an image. When a link, to change the system's state, requires the use of the camera, the captured bitmap is analyzed and if there is a Data Matrix code associated to that image, this code is sent to the Context Processor to manage the interaction accordingly.

4.1.1.3 Context Processor

This is the module of the engine that has to manage all the data gathered from the other components. This is the core module of the entire framework architecture, since the context has to be correctly detected in real time and the proper stimuli have to be presented at just the right time. At the start of the application, the general settings are loaded and the first frame becomes available to the users. Resources and settings are loaded from the stored XML data files. After loading a frame, the Context Processor has to manage user interaction which is set by the links. Furthermore, the context processor manages the use of every multimedia resource within the application. Context processor is aware of every defined link and, as one is activated, the next frame is loaded and the resources are shown to the users.

4.1.2 Authoring Tool Implementation

An authoring tool was developed allowing users to create new applications and to preview existing ones. While developing a new application or editing an already developed one, the authoring tool creates a set of files that will be used to provide the different settings to operate each module of the architecture. Mainly, those files are XML files that will be

[4] Hhttp://msdn.microsoft.com/en-us/library/ms783323(VS.85).aspx
[5] Hhttp://www.dtksoft.com/barreader.php
[6] Hhttp://barcodelib.com/net_barcode_reader/main.html

operated by the architecture when the application is executed within the framework. XML data was chosen since its use is generic and independent from any programming language. This way, this authoring tool can be replaced at any time by another that can produce the desired XML content. Furthermore, all these files obey to a structure which are defined by a XML Schemas and that is, therefore, an advantage to common files. Nonetheless, the creation of the XML files is achieved through a simple set of forms where the authoring tool collects information that will allow the framework engine to reason towards different frames, resources and links. To begin the creation or edition of the form menus, users have to define the applications general settings, shown in Figure 4.1 bellow.

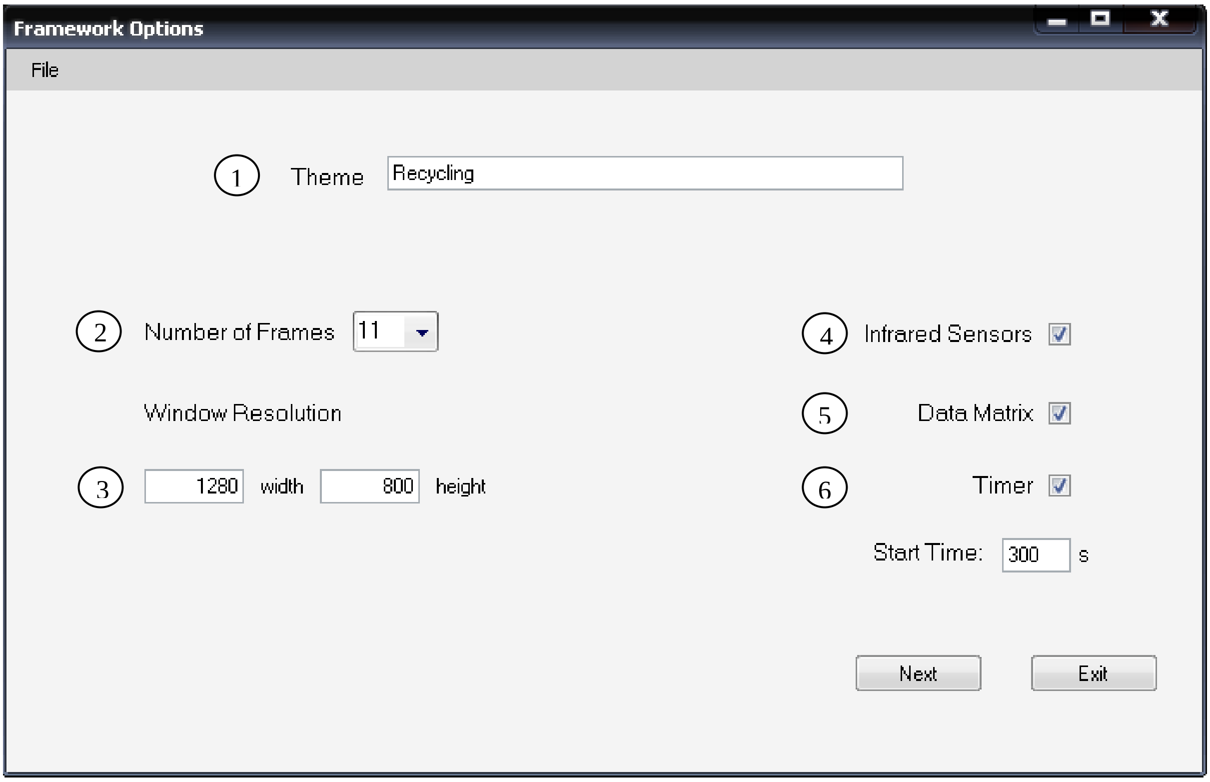

Figure 4.1 - Application's General Settings

(1) **Theme** - the theme is required and will identify the XML files for the application.

(2) **Number of frames** - the number of frames should be chosen according to the application's possible states.

(3) **Window Resolution** - this option is use to define the window resolution for the application. The default value is the screen's resolution.

(4) **Infrared Sensors**- This checkbox selects whether the application will use the infrared sensors or not.

(5) **Data Matrix**- This checkbox selects whether the application will use the Data Matrix images sensors or not.

(6) **Timer** - If this checkbox is selected, the user has to specify the period of time for clock countdown. The clock starts when the first frame with the “Show Timer” option selected is loaded. When the time reaches zero the system will reload the first frame.

The application’s general settings obey to a XML Schema that can be seen in Figure 4.2 bellow.

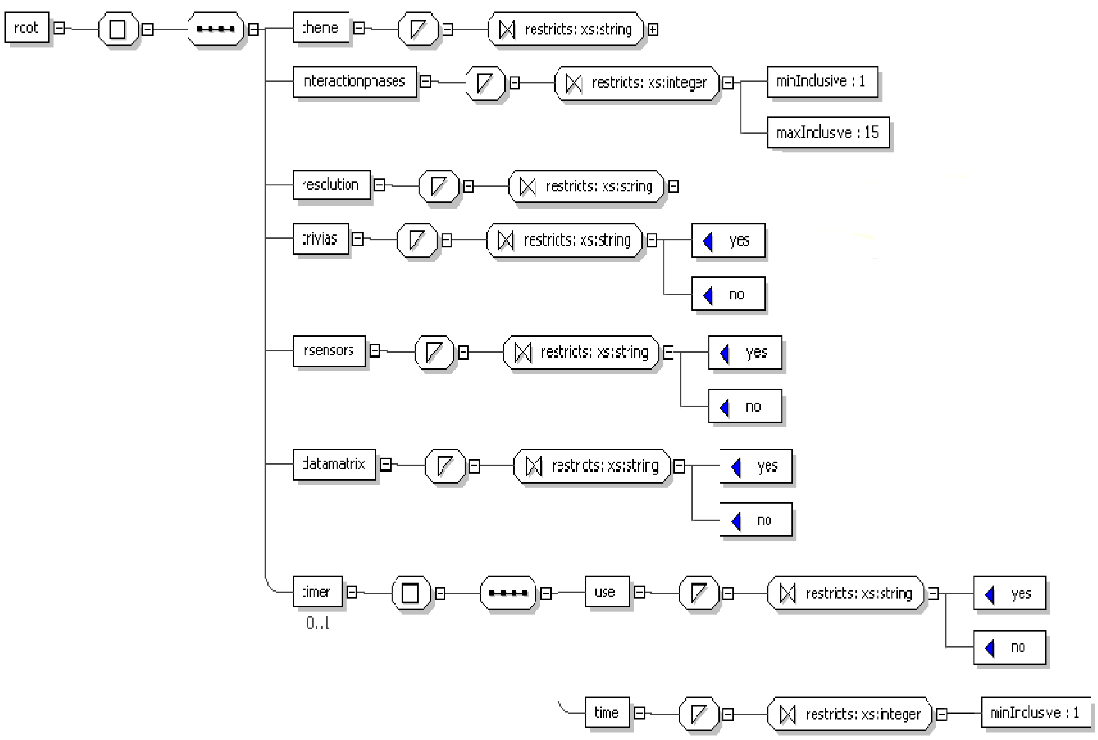

Figure 4.2 – Application’s General Settings XML Schema

A unique name is required for each application developed within the framework in order to create the structure to support it. Afterwards, the authoring tool provides a way to create the frames for the application as well as to define the resources and links associated with each frame. Frames correspond to application states and links establish the transition between different states. Although most persuasive applications are based on visual interfaces, the framework allows the creation of applications based on other stimuli, namely auditory stimuli. Thus, frame is a concept aggregating the stimuli supplied in an interaction point. When a specific context is sensed, a link is activated and a new frame is conveyed to the application user with the appropriate content in order to inform or persuade the users to adopt a certain procedure. Indeed, the flow of the framework is to detect a particular context, activate a frame and provide the correct stimuli. Figure 4.3 shows an example of a menu of the authoring tool for the creation of a frame.

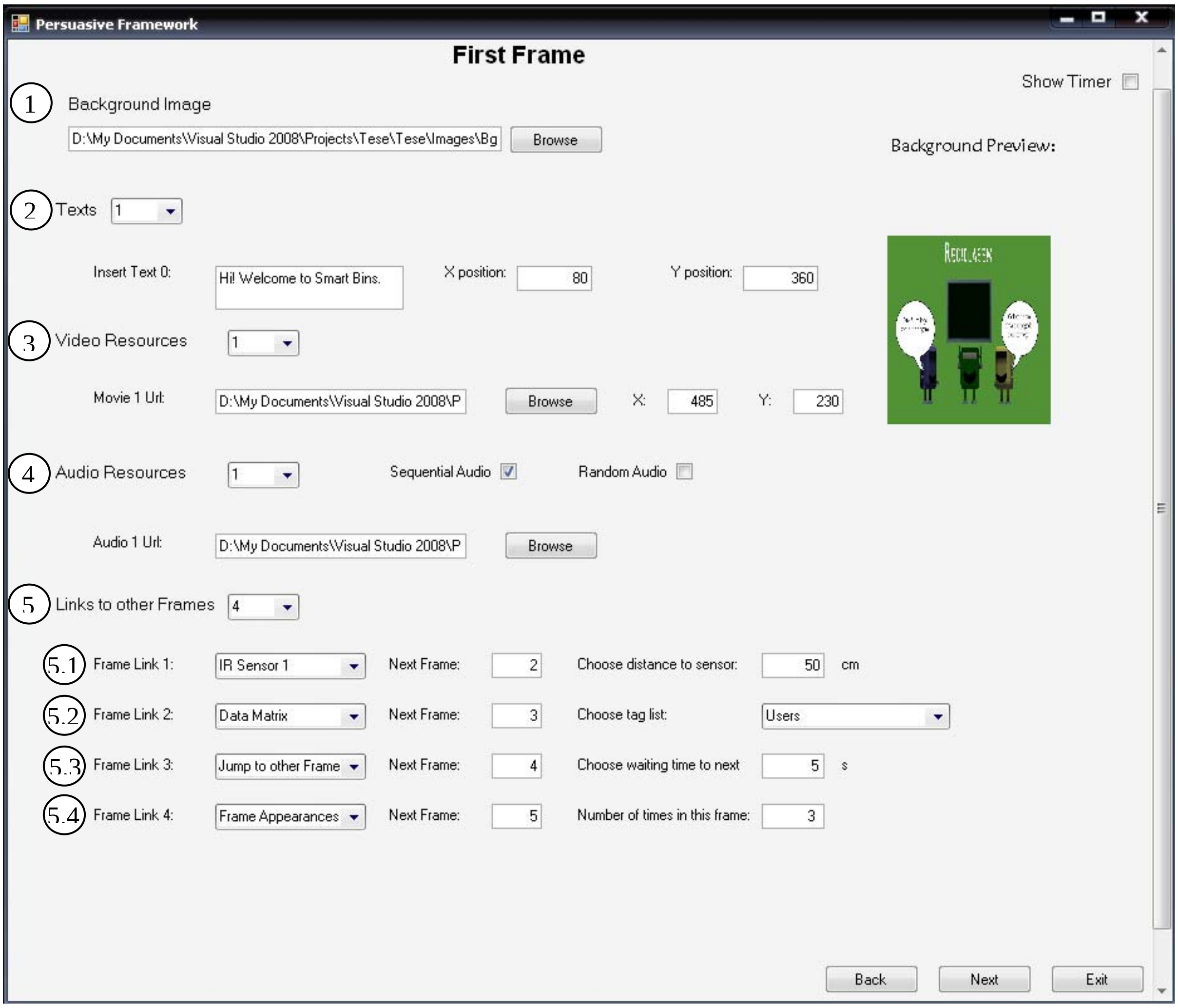

Figure 4.3 - Frame edition

(1) **Background Image** – if the application requires an image for the background, it should be specified here. Once chosen, a preview of that image is shown on the top-right side of the form.

(2) **Texts –** once required, users have to select the number of texts to appear on screen. According to that, and for each text element, users have to input the actual text and its position on the screen.

(3) **Video resources –** the number of video resources in a frame have to be selected in order to choose the appropriate video/s for the frame. The video position on the window is also required.

(4) **Audio Resources –** the audio resources may be presented in two different ways: as sequential audio or random audio (this is only for more than one audio resource per frame, if there is only one audio file associated to the frame, the audio starts to play when the frame is loaded).

- In the sequential audio mode, the audio files are played one after the other. The time between the reproductions of each audio resource may be also defined by the users.
- In the random audio mode, the audio files are played randomly each time the frame is loaded. Hence, there can be different audio stimuli for a frame.

(5) **Links to other frames** – the image shows an example of all the different types of links that may be chosen in the current implementation: the links from one frame to others can be, as explained before, defined by distance to the IR sensors, Data Matrix images, time in a frame, or number of times a certain frame occurs at runtime. All the links require the specification of the frame that will be loaded next.

(5.1) Infrared Sensors – when using the infrared sensors, the users have to provide the distance value, in centimetres, from which the system will activate the link and change the frame.

(5.2) Data Matrix – the use of the Data Matrix images requires the user to choose which list will be used for the frame change. If list is not already created the combo box allows the user to add a new list. This will lead to the Data Matrix edition menu which will be explained in detail further.

(5.3) Jump to other frame – when no user interaction is necessary to change frame, users can use this option. After specifying the frame to jump to, users have to choose the time, in seconds, for that change to occur.

(5.4) Frame Appearances – users may specify the number of times a frame occurs at runtime. Hence, if the frame is loaded the defined number of times, the system changes to the next chosen frame.

All the form menus for frame's edition have the same options, so the user is entitled to choose which resources to use in each frame, and which links will lead to other frames. All the frames can be different, but users may also choose the same resources and links for diverse frames. Also, the configuration files produced by the authoring tool for the application's frames obey to a XML Schema presented in Figure 4.4.

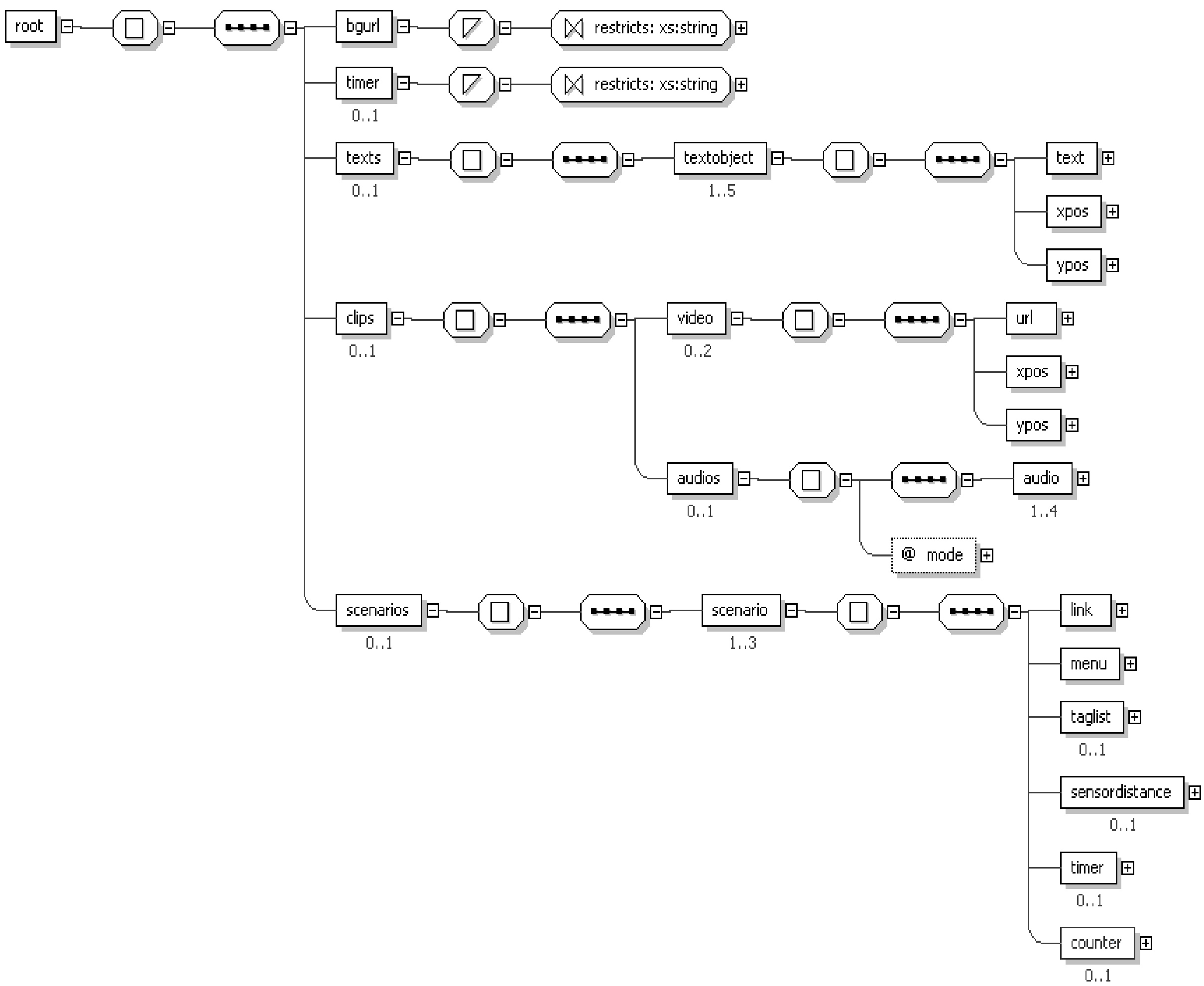

Figure 4.4 - Frame Settings XML Schema

Data Matrix links are a little bit different. Elements, which will correspond to Data Matrix images, have to be included in a list so the architecture may interpret which images can lead to other frames as different lists of images may link to different frames. This can be done with the help of the Data Matrix menu (Figure 4.5). For example, if one wants to produce an application where the users have to be identified with a card using a Data Matrix image, the names of the user to be identified have to be inserted in a list, so the framework can have information of which users can access the system.

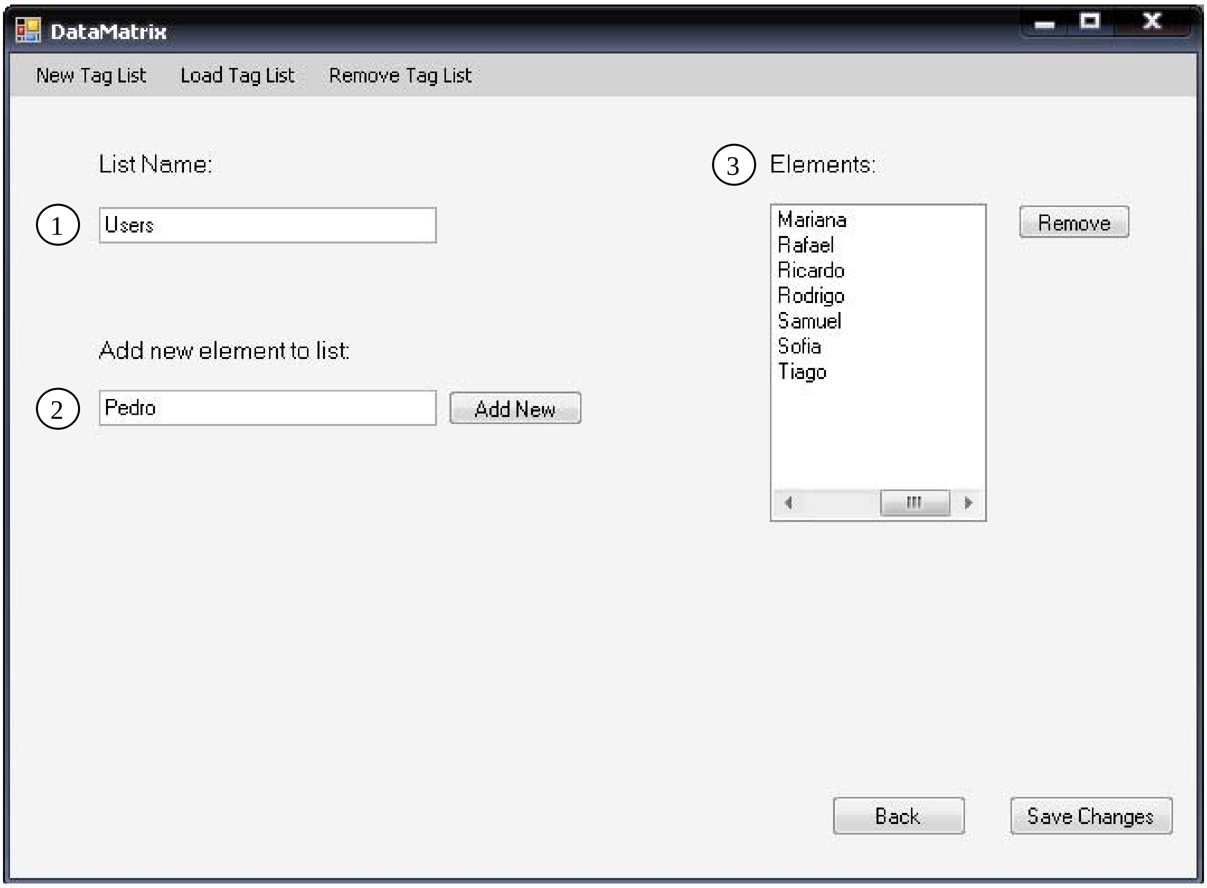

Figure 4.5 – Data Matrix list edition menu

This way, and taking the user's identification example, the Data Matrix list could be done as in Figure 4.4, where:

① **List Name –** this name will identify the list to be chosen in the frame menus, when defining a data matrix link, and also corresponds to the XML file for this list.

② **Add new element to list –** as the name states, this option allows the insertion of a new element in the list. The corresponding image is then downloaded from the kaywa[7] website and stored in a project folder for the user to be able to print it and use it when running the application.

③ **Elements –** this box stores and displays the elements present in a list. The elements can be removed at any time.

The XML Schema for the Data Matrix lists (Figure 4.6) obey to a simple Schema presented bellow.

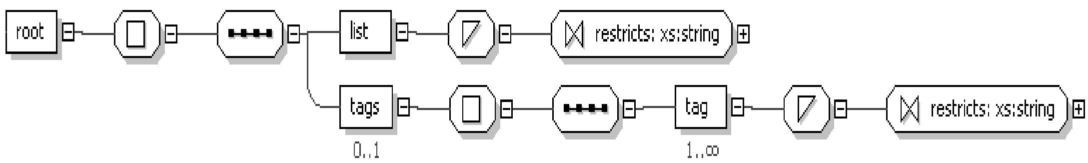

Figure 4.6 – Data Matrix lists XML Schema

Furthermore, in order to control the lists available for each application, the list names are also stored in a XML file. This file is stored in a Settings folder so that the engine can load the lists of tags associated with each application.

To support the authoring of new applications, in order to be easier to keep track of the application structure, it has been added a graphical support window that shows a graph representing the frames already created and the links between them. This way, the frame currently being edited by the application developer appears in the form of a red circle and the remaining ones are represented by green circles (Figure 4.7). This tool allows the user to visualize the current application structure, including the frames already created and the corresponding links established between them. Thus, users do not need to remember the

7 Hhttp://datamatrix.kaywa.com/

whole structure they have developed. Nevertheless, if the application developer realizes that he needs to make changes on the menus, it is easy to edit the previously saved XML files.

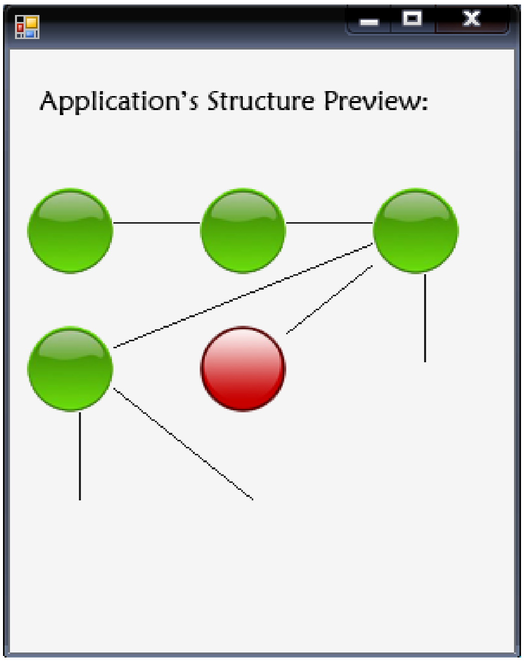

Figure 4.7 – Application's Structure Preview

The menus can be store and loaded at any time. Also, users are able to load any created application whenever they want to. This allows them make the appropriate changes until they reach the desired result. After configuring all the menus, the architecture has all the information necessary to build the application, including information regarding frames, resources and links.

4.2 Prototype Development

In order to test the presented framework, an application named Smart Bins was conceived and created with the objective of increasing young children's conscience towards recycling. The application had two main purposes: test the framework's authoring tool usefulness and usability to develop new applications and explore the engine's capabilities to create context aware environments. Smart Bins was designed as an augmented recycle bin, composed by three containers (blue, yellow and green), able to detect if users are placing the objects to be recycled in the correct container. Smart Bins is also able to communicate simple persuasive messages right on the appropriate time. For the development of Smart Bins, a set of sensors, composed by four infrared sensors was used for measuring distances. Multimedia content (videos, images, sounds) used in the Smart Bins development was collected and prepared in advance. This application had children as target users, therefore, to motivate them, the application had to be amusing and, at the same time, challenging. Having this in mind, one realized that the best way to reach this solution had only one way to follow: the creation of a game. This game had to follow as many persuasive concepts as possible and still be entertaining. The idea was to guide the users to an experience, providing stimuli according to their interaction and present them with useful information about the subject in hand. This way, the images for the menus were created and the texts were chosen to convey these intentions to the users.

Smart Bins was then designed as a game intended to change children behaviour towards environmental issues, namely recycling. Target users are children with ages between from 7 to 10 years old. This initiative intended to approach a serious problem, such as the necessity for recycling, trying to make recycling a funny activity and at the same time teaching children how to do it correctly and showing them why they should recycle. The proposed application can address this issue in an entertaining way, also enabling children to earn conscience on this subject. Mascots were also created as an anchor throughout the entire application, guiding users and providing them with useful stimuli. The mascots, resembling humanized recycle containers, are displayed on the appropriate application frames to convey persuasive messages to the children, bringing "life" and intelligence to Smart Bins. Figure 4.8 shows a preview of Smart Bins and the introduced mascots. Smart Bins works as follows. When no one is close to the system, the display shows a video about recycling. As soon as someone approaches the augmented recycle bin (detected by one of the infrared sensors), the application changes state and one of the recycle bins characters, tells the children how to play the game and start recycling.

To play Smart Bins, children have to place a set of objects in the recycle containers, aiming at selecting the correct container for each object. A Data Matrix image is attached to each object to identify the type of container in which the object has to be placed (green, yellow or blue). Thus, to identify an object the user has to place the object in front of the camera and then place it the recycle container. The system tells the user if they made a correct action. Figure 4.8 shows Smart Bin's stand-by menu image.

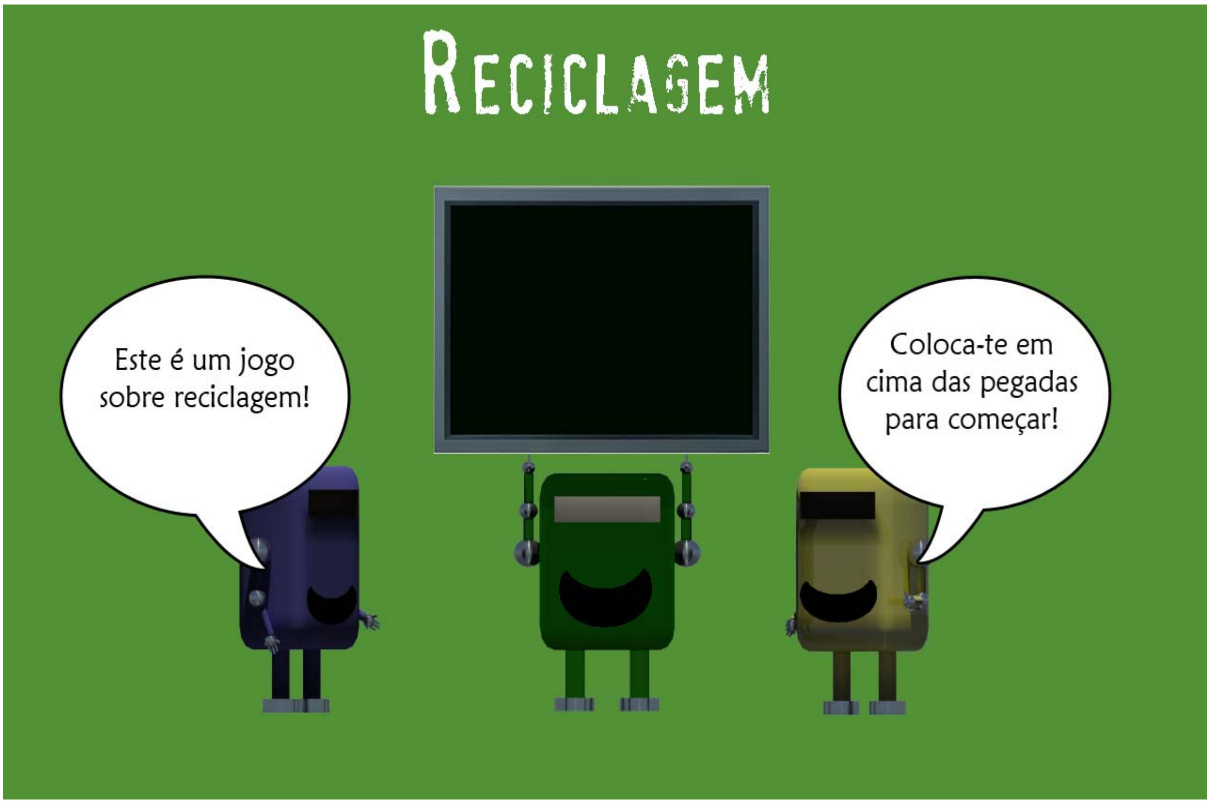

Figure 4.8 - Smart Bins stand-by menu

Each container has an infrared sensor, thus whenever an object is put inside a container, the object is detected and stimuli are provided to children telling them whether the object was placed in the correct container. A correct identification of the recycling container for the object links Smart Bins to a congratulation frame with audio to reinforce the success of the user's action. Otherwise, when children fail to identify an object, Smart Bins mascots get sad and encourage them to do it better with the next object. Smart Bins was set to end after a period of time, if users failed to identify more than three (3) objects or after the recycling of a predefined number of objects. For testing purposes, the application game goes on for five (5) minutes, encouraging children to correctly identify seven (7) objects. When children identify all the objects, or the time ends, results are displayed and the system conveys messages about the user's actions results. Besides the score, objects correctly and incorrectly recycled, the game provided interesting facts about recycling. These texts were carefully selected from the Sociedade Ponto Verde website [8]. Smart Bins explores the use of almost every one of this framework's components, except flash animations, as it uses all the remaining modules available for this implementation.

[8] Hhttp://www.pontoverde.pt/

5. Evaluation and User Studies

5.1 Evaluation with User Participation

As it is not easy to involve users in the design process due to lack of resources and time, a usual approach is to evaluate the applications through user participation. As there was not a possibility to involve users in the design process, the application was only evaluated after the first implementation stage. Given this, it is important to discuss the evaluation techniques and present the approach followed. Alan Dix et al [22] presented a study of evaluation techniques which is summarized in the tables below.

Table 5.1 - Classification of experimental and query evaluation techniques [22]

	Experiment	Interviews	Questionnaire
Stage	Throughout	Throughout	Throughout
Style	Lab	Lab/Field	Lab/Field
Objective	Yes	No	No
Measure	Quantitative	Qualitative/ Quantitative	Qualitative/ Quantitative
Information	Low/High Level	High Level	High Level
Immediacy	Yes	No	No
Intrusive	Yes	No	No
Time	High	Low	Low
Equipment	Medium	Low	Low
Expertise	Medium	Low	Low

Table 5.2 - Classification of observational evaluation techniques, based on [22]

	Think aloud	Protocol Analysis	Post-task walkthrough
Stage	Implementation	Throughout	Throughout
Style	Lab/Field	Lab/Field	Lab/Field
Objective	No	No	No
Measure	Qualitative	Qualitative	Qualitative
Information	High/ Low Level	High/ Low Level	High/ Low Level
Immediacy	Yes	No	No
Intrusive	Yes	No	No
Time	High	Medium	Medium
Equipment	Low	Low	Low
Expertise	Medium	Low	Medium

Framework Evaluation:

As one needed non-intrusive, low time consuming and low equipment techniques, the chosen technique for the framework evaluation, at the query level, was the questionnaire. This way, users could evaluate the creation of an application using this framework and express their likes, dislikes and opinions when filling the questionnaire. Protocol analysis using paper and pencil to register user's steps was chosen at the observational level for the same reasons. The think aloud method was not followed to avoid distracting users from their tasks, since user tasks demanded for a high level of concentration This way, while users were testing the framework, an observer was annotating their actions and performance. At the end of each user test, the user had to answer a questionnaire related to their experience using the framework.

Smart Bins Evaluation:

The evaluation methods for smart bins had to be a little bit different as one had children evaluating the application. When dealing with children, in planning and delivering any survey, there are three aspects where attention is needed. The first is the consideration of the sample (group of users), the second is the mode of questioning and the third are the questions themselves [26]. The sample is normally a convenience one. The sample selected for these tests was a single children's class. When designing the questions for a survey there are four general principles that should be considered:

1. Ask for the minimum of information required for the purpose
2. Make sure questions can be answered
3. Make sure questions will be answered truthfully
4. Minimize questions that will be unanswered

These principles were taken in consideration for the test's questionnaires. One also used the Smileyometer [26] in the questionnaires to evaluate user's motivation and satisfaction. The chosen query technique was the interview, as in comparison with the questionnaire, this technique brings two main advantages. First, at the chosen age, children are not particularly comfortable reading questionnaires, even if the questions are simple, and so they could get easily bored and simply ignore the questions. Second, the person who is registering children's answers can also register their reactions and better evaluate the truthfulness of those. As the tasks were evaluated while children were playing a game, it would be difficult to use the think aloud method, nevertheless their observations were annotated by an observer. Post-task walkthrough observational techniques were not followed, since it is hard for children recall and describe the interaction that they had during the game. Hence, one chose the protocol analysis as the observational technique. Children were interviewed before the test, then they played the Smart Bins while an observer registered the notes on their actions and, after the test, they were also interviewed regarding their experience. Both interviews were guide by a questionnaire.

5.2 Framework Studies

The framework was subjected to usability testing, aiming at evaluating its complexity, usefulness, and understanding on how easy it was to learn and use. We also intent to establish what further improvements should be implemented in future development stages to increase usability and usefulness. Participants were asked to use this framework to create the Smart Bins application according to the instructions given in the form of a storyboard (Annex A). The storyboard explains the multimedia resources and the links they should use to create each interaction frame, as well as the general idea of each component that participants could configure to use. We intended to find out if the users understood the framework functionalities and if they could reach the expected result, successfully developing Smart Bins. The tests were divided in four parts:

- An introductory questionnaire to understand if users knew the concepts of ubiquity and persuasive systems, if they were regular computer users and if they knew how form-based menus worked.
- Briefing of the framework and the objectives of the test, explaining how Smart Bins should work, and pointing out the multimedia contents users could use and the way they could define the transitions between the frames.
- Application development by filling the system's forms according to the storyboard of the application.
- A final questionnaire to evaluate the potential of the framework and to understand user's opinions concerning its usage and usability of the authoring tool when creating the persuasive application.

After the introductory questionnaire and the briefing, users started to develop the proposed persuasive application – Smart Bins – according to the storyboard they received. To create Smart Bins, participants had to perform a series of 74 simple tasks. These tasks included, for instance, choosing the background image and all the multimedia resources for each interaction frame, and also defining the links that establish transitions from one frame to the others. During this part of the test, an observer watched the users and registered their performance in each of the actions. The tests were carried out individually.

The users' performance in each task was scaled from 0 to 3, where:

- 0 – Not completed tasks
- 1 - Incorrectly completed tasks
- 2 - Tasks completed with help or not easily completed
- 3 - Easily completed tasks.

5.2.1 Participants

The tests were performed on a population of six (6) voluntary participants, four (4) of them male, and two (2) of them female. All of them were college students, from different study areas as Physics, Biomedical and Computer Science, with ages between 23 and 30 years old, yielding an average age of 25.3 years old. All the participants had their first contact with the framework during the test.

5.2.2 Questionnaire

Besides gender and age, the questionnaire covered general data on users' knowledge about persuasive and ubiquitous systems, as well as their aptitude to work with a computer and with forms. To conclude the tests, the participants had to complete a nine questions questionnaire. Each participant was presented with two questions to rate how easy it was to build Smart Bins using the framework and the overall impression of the framework. To answer these questions, users had to circle a response on a 5-point Likert-type scale. On this scale a response of 1 (one) means very poor and a response of 5 (five) very good. The remaining questions were open-ended and covered participants' opinions on the potential use for this framework for the creation of persuasive context-sensitive applications, different from the application they just created, as well as their likes and dislikes about the framework and their ideas on how to improve it.

5.2.3 Results

Regarding the introductory questionnaire we noticed that two (2) of the six (6) users knew the concept of ubiquity, four (4) had a general idea of the meaning of persuasive technology, and they were all regular computer users and also had previous contact with forms. Table 5.3 shows the results of the tasks performed by the participants to configure the XML files, using the authoring tool, which resulted in Smart Bins application. For each participant, considering the scale presented above in this section, we counted how many actions were accomplished within each level of performance.

Table 5.3 – Participants tasks performance results

	0	1	2	3
Participant 1	0	1	2	71
Participant 2	0	10	0	64
Participant 3	1	0	3	70
Participant 4	0	0	2	72
Participant 5	0	0	1	73
Participant 6	0	0	4	70
Evaluation Count	1	11	12	420

The performance of participants during the executed tasks was very satisfactory. Users understood quite well how to create the application using the authoring tool menus, as most of the tasks were easily completed (Mean = 2.92, Standard Deviation = 0.24). To better understand these results Figure 5.1, on the next page, illustrates the participants' results during Smart Bins creation.

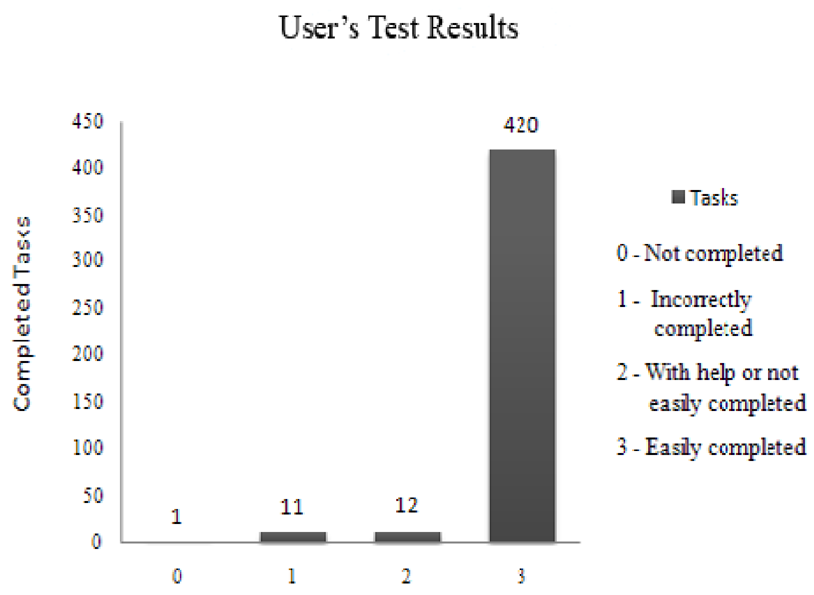

Figure 5.1 – Test results

Regarding the final questionnaire, participants said that it was easy to create the application using the authoring tool (Mean = 4.00, Standard Deviation = 0.57) and they revealed a good overall impression about the framework (Mean = 4.3, Standard Deviation = 0.97). Regarding the third question, which asked if participants felt they could create different persuasive applications using the framework, the answer was unanimous. All participants answered that they could use the framework to create other context-sensitive persuasive applications. They even suggested some examples, such as an application aimed at improving eating habits, or another one to enhance learning methods. Concerning what they liked best about the framework, the answers were diverse, and included the simplicity of the framework to create new persuasive applications, its versatility, and the fact that during the edition of the Smart Bins application, they could observe the evolution of the application structure (as a graph) as they create new interaction frames and establish the links between them. The weakest point of the framework identified by the users was the limited help functionalities available to assist especially new users.

Users had the opportunity to make suggestions to improve the framework. All participants agreed that the interface to preview the application's structure should be improved. Half of the participants suggested the possibility to add flash animations to interaction frames. This feature was later included in the framework as described in chapter 3. One participant wanted to have a way to respond to sound inputs. Finally, when asked about the persons who could use this framework to create new persuasive applications, participants said this framework wasn't accessible to everyone, but people with good logic reasoning should have no problem in using it. In general, the participants understood the functionalities of the framework and considered it useful and easy to use. The tests' duration was from 30 minutes to 1 hour depending on the users' understanding of how the Smart Bins should work. The users found that the hardest part to accomplish was the establishment of the transitions between interaction frames. That may have been caused, at least partially, by the fact that the application they were trying to build had not been designed by them.

5.3 Smart Bins Studies

We also wanted to test the Smart Bins application developed with the framework, in order to assess its usability and persuasion effects. A series of two user tests sessions were performed with children in a school. Both sessions had the same participants and the second one was carried out one month after the first one. The objective was to understand how users would adapt to this new type of interaction and to assess the short and mid-term effect the application would have on the users' behaviour. At the beginning of the tests, users where given a brief explanation about this game's objective and they started to play right after. The two user tests sessions were identical (same participants, same applications and game settings), in order to minimize the external variables that could influence the results. As in the framework tests, these tests were divided in four parts:

- **Introductory questionnaire –** the purpose of this questionnaire was to evaluate children's motivation to play the game and the importance they assigned to recycling activities.

- **Brief explanation –** one briefly explained how to play the game. Users where told that they had to follow the instructions given by the game. Children also had to identify the objects with the use of the camera and to insert 7 fixed objects in the correct containers. Children had 3 more possible objects to choose from if they failed any of the fixed objects. It was also explained that the game had to be completed in 5 minutes.

- **Interaction analysis –** Users were observed while doing four different tasks: stand in the right place to start the interaction; identify themselves to the camera; identify an object and put an object in a container. All the interaction tasks were analyzed and the users' performance was rated on a scale described below, depending on its success.

- **Final questionnaire –** when participants concluded the game, they had to answer a set of questions to register their feedback on the experience.

During interaction analysis, an observer watched the users and registered their performance in each of the actions. The tests were carried out individually.

Again, the users' performance in each task was scaled from 0 to 3, where:

- 0 – Not completed tasks
- 1 - Incorrectly completed tasks
- 2 - Tasks completed with help or not easily completed
- 3 - Easily completed tasks.

5.3.1 Participants

Seventeen (16) children participated in both tests, eight (8) male and eight (8) female, with ages between eight (8) and nine (9) years old with average age of 8.56. The first test had one more participant than the second. As stated before, the participants were the same in both test sessions (except for one participant that missed the second test). They all had their first contact with the application in the day of the first test. The tests took place at their school so it would be a familiar environment and they would feel more comfortable.

5.3.2 Questionnaire

The questionnaires were a little bit different in the two occasions. In the first experience, the type of questions were meant to evaluate the interaction and less the effect of the game in the children's behavior towards recycling. On the other hand, the second test was directed to the persuasion effects of Smart Bins. Besides name, age, gender and motivation level, (rated from 1 to 5, where 1 stands for highly unmotivated, and 5 stands for highly motivated) in the first test's introductory questionnaire users answered if they use to recycle regularly. On the other hand, the second test's introductory questions asked users if they had f recycled more often since the first test and if they had talked to anyone about the game. The final questionnaire covered, in both cases, user satisfaction after playing the game. In addition, in the first tests, the questionnaire had six open-ended questions. Children could tell if they thought that the game was easy to play, if they would like to play it more times, if they wished to have the game in other places besides their school, if they felt they have learned about recycling activities and would recycle more in the future. In the second day, with the final questionnaire one intended to understand if the children maintained the opinions given in the first day. Furthermore, one wanted to know if they felt more comfortable playing the game in the second occasion or in the first and if they had recycled more since their first contact with the application during the first test.

5.3.3 Results

First test:

In the universe of the seventeen (17) children that participated in these tests, the introductory questionnaire on the first day revealed that ten (10) children recycled now and then at home, three (3) of which also recycled at school. This means that seven (7) children never recycle at all and that none of them had recycling habits. Their motivation to play Smart Bins was scaled from 1 to 5, as stated before in section 5.2.2, and revealed that they were excited to play the game (Mean = 4.00, Standard Deviation = 0.97). Regarding the analysis on their interaction, the graphics bellow show how users responded to the proposed tasks.

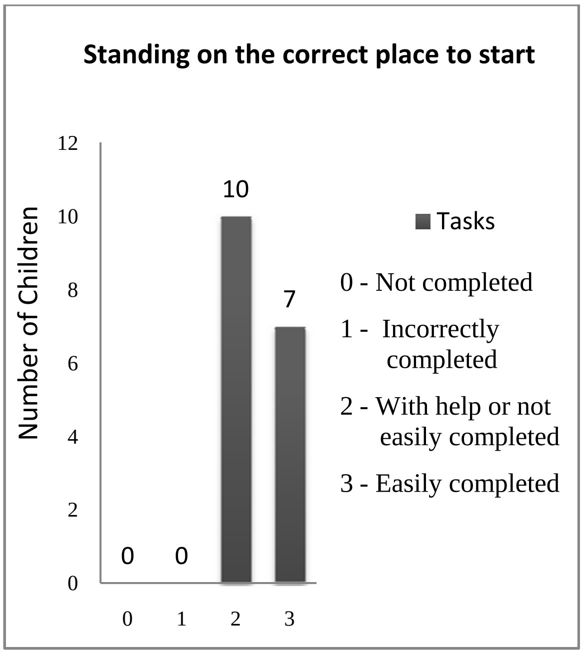

Figure 5.2 – Users that stood on the right place

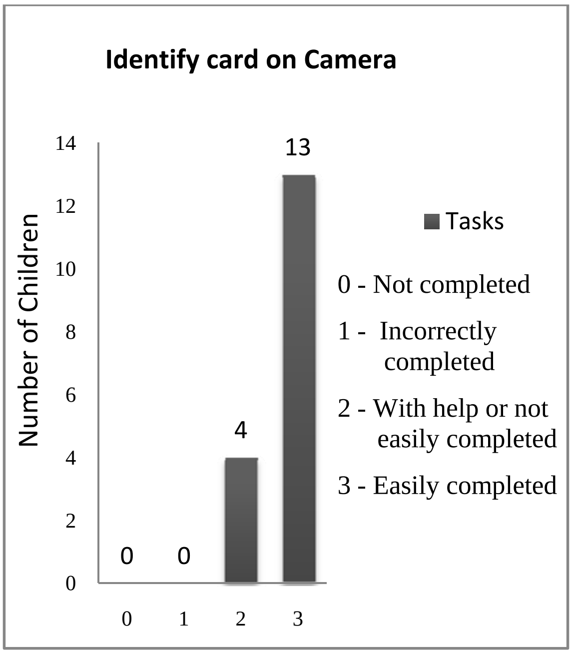

Figure 5.3– Users that identified card on the camera

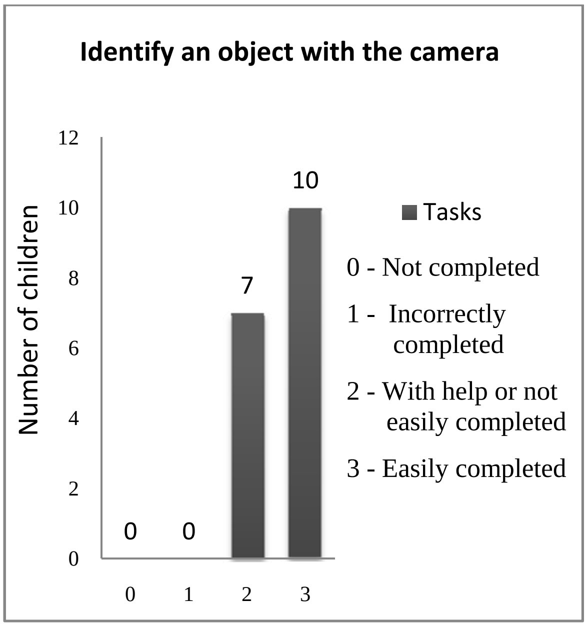

Figure 5.4 – Users that identified the objects

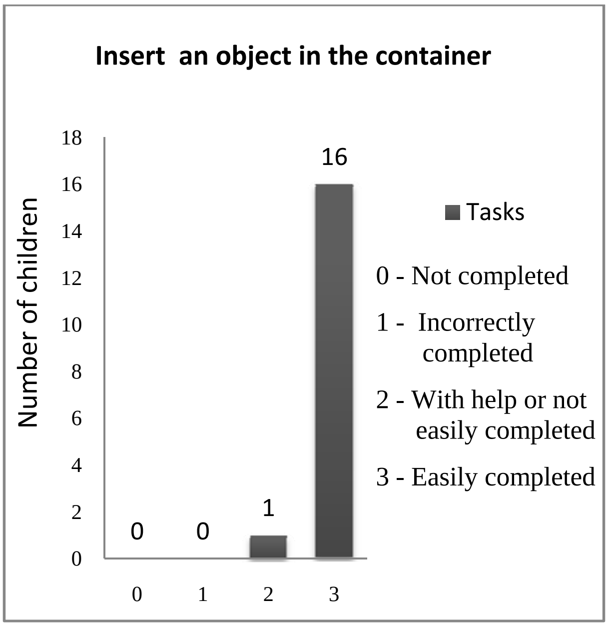

Figure 5.5 – Users inserting objects in the containers

In the first tests five (5) of the children placed all the objects correctly, six (6) of them inserted one (1) object in the wrong container, three (3) of them missed two (2) objects and three (3) missed three (3) objects. Only one (1) user was not able to finish the game.

Regarding the final questionnaire, one evaluated users' satisfaction after playing the game and realized that they appreciated the experience. On a scale from one (1) to five (5), children revealed that they were very satisfied (Mean = 4.71, Standard Deviation = 0.57). Concerning the final open-end questions, twelve children thought that the game was easy to interact with and five of them answered it was neither easy nor difficult. When they had to answer if they would like to play again, the answer was almost unanimous, only one user was not sure of that, all the other wanted to repeat the experience. Fourteen (14) of the players wanted to see Smart Bins available in other places, three (3) of them said they didn't and one wasn't sure about it. Nevertheless, they all agreed that they would like to have this system at home. Thirteen (13) interviewees said they learned about recycling with this experience and four of them said they did not. Finally, all the children that said, before the test started, that they weren't used to recycle shared the same opinion and felt that they would recycle more after that day.

In general one noticed that users did not realize that they had to approach the containers to start playing the game, mainly because it was not obvious. The sensor only detected the children presence when they were very close to the system and the children did not paid much attention on the screen messages. Moreover, users did not realize that some frames changes during the interaction, mainly when there was not audio involved signalling it. Moreover, the display was situated a little bit far from the camera which difficult their vision of the screen changes. Since most of the children did not read the game instructions on the screen, some users did not realize that they had to identify the object on the camera before inserting them in the containers and they had to be warned when dealing with the first object. This revealed the importance of the audio instructions which were already implemented to cover the cases of children with more reading difficulties. The sounds were not easy to understand due to its quality and the surrounding noise. At some point, the lighting conditions influenced negatively the objects identification and that lead to a setback in the interaction. The children failed to choose the correct container on a total of twenty-one (21) objects with average of 1.24 failed objects per child. One curious fact: the object that was placed more times in the wrong container was the milk package, followed by the cereal bar package.

In conclusion, Smart Bins seems to be easy to use and the user's opinions made us think that they have been sensitized about recycling and, at least part of them will be more concerned about this matter.

Second test:

In this second day of testing, one intended to analyse how users felt about recycling, assessing if there was any changes in their use of the application or in their attitude towards recycling. One wanted to find out if there was any signal of change in their attitude and in their way of thinking that lead to a behaviour change towards recycling. One month after the first tests, all users recalled their first day of testing and they all were very excited to play again. Sixteen (16) children's tests were taken into account. This was one less test than before because one of the children could not be present. The photography bellow (Figure 5.6) was taken during the tests and shows one of the children playing with Smart Bins during the tests.

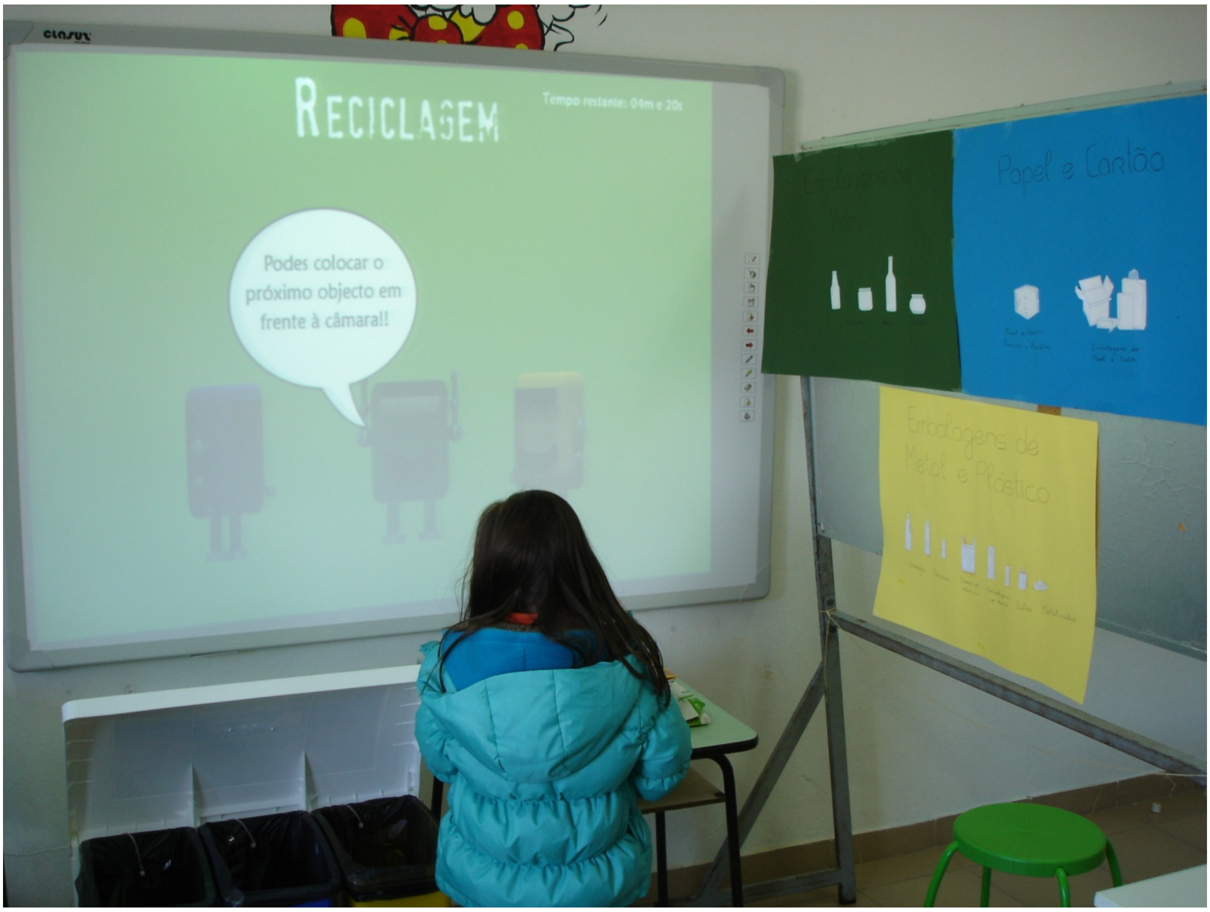

Figure 5.6 – Smart Bins Environment

In the introductory questionnaire, one wanted to know if any of them felt that they had recycled more after their first experience with Smart Bins. Six (6) interviewees answered that they did not, one (1) said that he had done a little more and the remaining nine (9) said they had recycled more from that day on. Six (6) children recycled more at school, one (1) at home and in school, one (1) just at school and finally one (1) of the children said that she now recycles at her parents bar. Only one participant answered that he did not talk to anyone about the game but the other fifteen (15) did. Generally people asked them if they enjoyed the experience, if they had learned about recycling and their answers were positive. When asked about their motivation to play again, the results show that they were really excited (Mean = 4.94, Standard Deviation = 0.24), again in a scale from 1 to 5. The tasks to be performed by the users were the same as the ones performed during the first day of tests and only small changes were performed in order to improve the interactivity. The users still had to identify and place in the correct container, seven (7) out of ten (10) objects in five (5) minutes. Their tasks were evaluated by an observer and the results are presented in the following graphics.

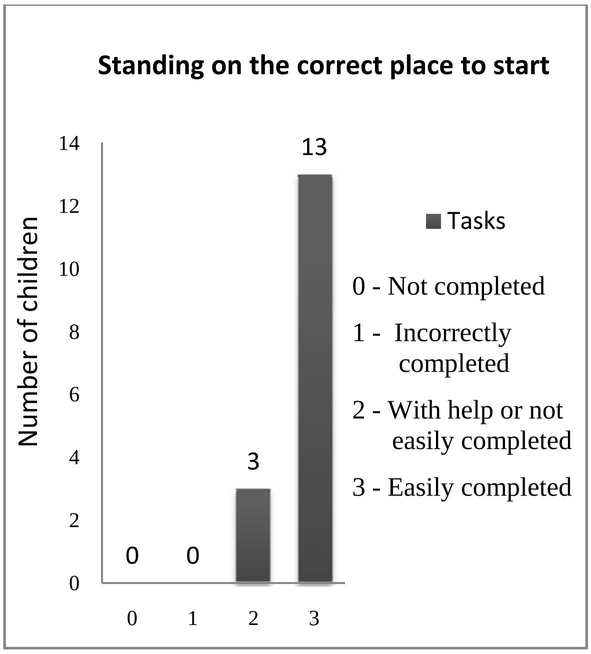

Figure 5.7 – Users that stood on the right place (2nd day)

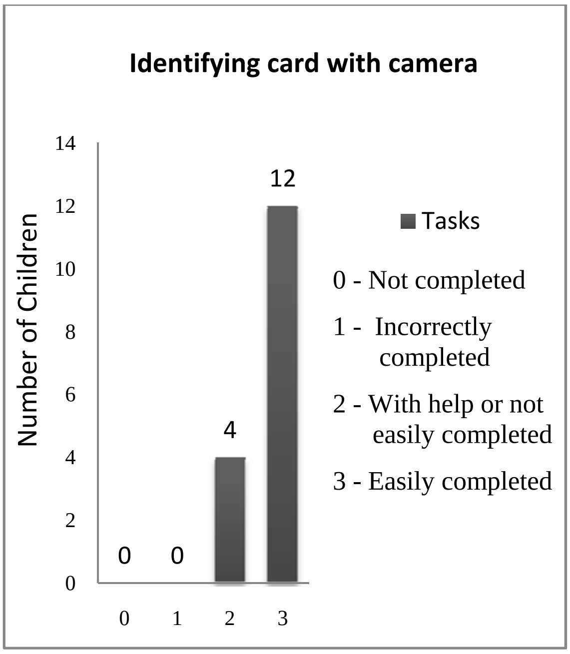

Figure 5.8 – Users that identified card on camera (2nd day)

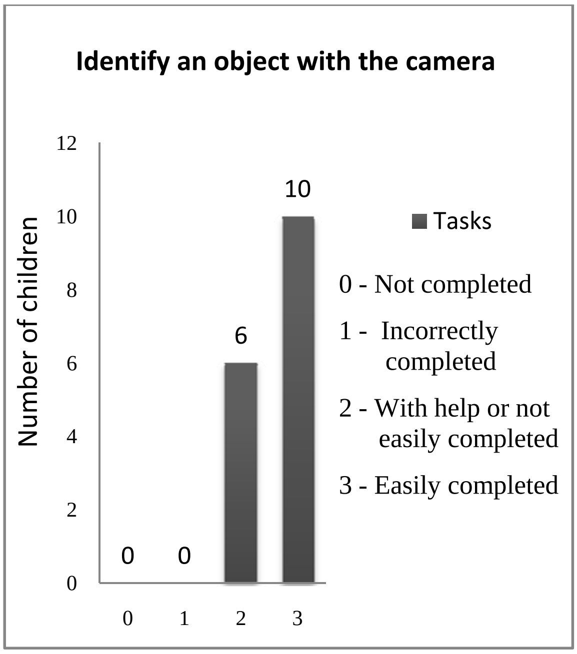

Figure 5.9 – Users that correctly identified objects with the camera (2nd day)

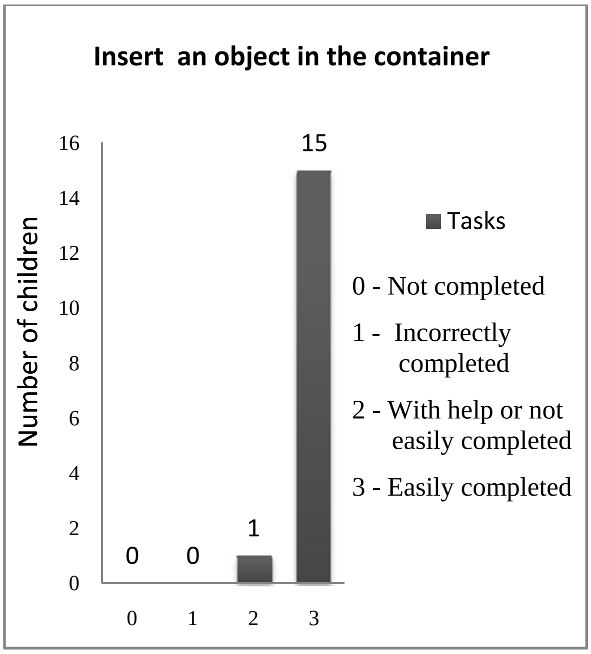

Figure 5.10 – Users that inserted the objects in the container (2nd day)

Only one user was not able to finish the game, which was the same user who could not finish it either in the first day. Three (3) participants finished without any wrong object placed, the majority (9 participants) failed to place one (1) object in the container and four (4) failed to place two objects. Regarding the final questionnaire, almost everyone was very satisfied after playing the game (Mean = 4.88, Standard Deviation = 0.33). They all agreed that they had learned about recycling. Five (5) participants answered that they preferred to play in the first day, mainly because the type of game was new and two participants felt the game was easier to play in the first experience; seven (7) participants liked the second day best because they thought the interaction was easier and one participant because he had learned more; the remaining four saw no difference and liked both times equal. They all felt they would recycle more if they had the game at home or at school and, finally, eleven (11) users said they would recycle more after these experiences and five (5) weren't sure.

With this second test one concluded that there was not a significant evolution in the usability due mostly to the fact that the results had been good in the first tests. Nevertheless, the children were more familiarized with the interaction and they completed their tasks faster. This indicated that the Smart Bins is easy to learn and use.

There was a difference in the performance of the first task mainly because an audio explanation was added to the frame in question. In the first day of testing there was not any audio support to tell the children to approach the containers to start playing the game and in the second day of tests that reminder made the difference. The children failed to choose the correct container on a total of seventeen (17) objects with average value of 1.06 failed objects per child, which represents a slightly positive evolution in their recycling skills, although it was a significant change. In order to significantly improve recycling skills and stimulate considerable behaviour change, the Smart Bins must be used in a more continuous manner.

As in the first day of testing, the milk package was the object that users failed to identify the most. The cereal bar package was the next object with more incorrect collocations. Both objects were treated as paper objects while they should be recycled in the plastic and metal container. One could also observe that the children involved in the tests did not pay attention to textual instructions. The audio was, therefore, a good solution to support the interaction.

6. Conclusions and Future Work

Context-sensitive systems and augmented physical objects are increasingly being used for persuasion purposes. As technology advances into the future being present in everybody's life, it is only logic to adapt it to humans, as much as possible, instead of the opposite. Moreover, if technology is used for so many purposes it should also be used to help people to adopt healthier behaviours. When it comes to persuasion, computers have many advantages over humans, which ought to be used to provide healthier life styles.

This work merges ubiquitous computing and persuasive technologies for the establishment of new forms of interaction with the users. The main objective is to allow a persuasive interactive system to capture the appropriate data concerning the users and their surrounding environment, and non-intrusively persuade them to adopt environmentally sustainable behaviours. In the scope of this master thesis, a framework was developed to allow the authoring of new persuasive ambiences which, based on the presence of different sensors throughout the environment, capture contextual information and produce the appropriate feedback. This framework was implemented following the underlying principles of persuasive technology such as tunnelling, tailoring, suggestion, surveillance and conditioning. Also, the use of multimedia resources, namely images, videos and flash animations, allows the developed applications to be seen as social actors. An authoring tool was developed in order to provide a simple way for users to develop innovative and different persuasive applications using the framework.

User tests were performed in order to assess the easiness of building a contextual aware persuasive application using the authoring tool. As the tests indicate, the authoring tool is simple to use and participants easily assimilated the concepts of frames, links and resources, which are the main concepts for the development of these applications. As users suggested, the graphical interface preview for the creation of new applications should be improved and a user guide is also scheduled for future work.

Another goal to be achieved is the development of a new mechanism to facilitate the incorporation of new types of sensors. This mechanism will have the same support of an authoring tool making it easier to create the configuration files necessary for each new sensor type. Furthermore, the suggestion of allowing the integration of flash animations had already been taken into consideration and was integrated on the current framework's engine.

Various multimedia resources have already been considered, however it is easy to integrate new modules to manage other resources therefore increasing the value and completeness of this framework. Although it has not been implemented yet, the framework should suggest, to application's authors, trustworthy information provided by third parties in order to add credibility to future applications developed with it. Furthermore, new integrated sensors should aim at recognizing further users' characteristics in order to broaden the group of target users to be persuaded. This would also extend the completeness of persuasive concepts covered by this framework.

Additionally, a prototype application - Smart Bins - was created with the use of the framework. Smart Bins aimed at promoting the recycling of waste materials, preventing greater future damages to the environment. Furthermore, with the purpose of evaluating Smart Bins' persuasiveness effect and usability, one performed a series of two tests with children. The first test measured their adaptation towards the system, in terms of usability, and considered the motivation and satisfaction before, during and after playing this game.

After this first test the application's stimuli was slightly improved and a month later another test was performed. There was no need for considerable enhancements, since the interaction had been very satisfactory during the first test. The second evaluation had, once again, the main purpose of testing the persuasive game in terms of usability and user motivation, but this time the persuasion effects and the educational capabilities of Smart Bins were also evaluated.

The final results were better than expected. Although the users have only played the game twice, the results show that most of them learned about recycling. Furthermore, there was a significant change in the way they considered recycling. Moreover, 56% of the children said that they had recycled more since they played the game for the first time. For a more detailed and meaningful analysis of Smart Bins' persuasiveness, it would be very interesting to test its persuasive effects in a continuous way.

Finally, one concludes that this framework can produce augmented environments that lead users to at least think twice on the subject in hand, which is a good indicator for persuasive purposes. This study is a starting point to develop new attempts to try to change people's behaviours towards the adoption of healthier life styles. Having in mind all the considerations deemed in this chapter, and taking into account how users can easily build new persuasive environments using this framework, it is expected that future applications may be even more useful. But at least one, Smart Bins, has already done a small contribution for a healthier environment.

References

[1] Romão, T., Hipólito, J., Correia, N., Câmara, A. & Danado, J. *Interactive Public Ambient Displays for Environmental Persuasion*, 2nd International Conference on Persuasive Technology (Persuasive 2007), Stanford University, Palo Alto, CA, 26-27 April, 2007.

[2] Fogg, B.J. *Persuasive Technology – Using Computers to Change What We Think and Do*. Morgan Kaufman, San Francisco, CA, 2003.

[3] Harjumaa, M. & Oinas-Kukkonen, H. *Persuasion Theories and IT Design*. In: Lecture Notes in Computer Science, Persuasive 2007, eds. Y. de Kort et al., Springer-Verlag, Berlin Heidelberg, pp. 311-314, 2007.

[4] Fogg, B.J. *Persuasive Computers – Perspectives and Research directions.* Proceedings of the Conference on Computer-Human Interaction (CHI), pp. 225-232, 1998.

[5] Andrés del Valle, A.C. & Opalach, A. *The Persuasive Mirror: computerized persuasion for healthy living*. Procedings of Persuasive, Eindhoven, Netherlands, 2006.

[6] Intille, S. S. *Ubiquitous computing technology for just-in-time motivation of behavior change*. Proceedings of Medinfo. vol. 11(Pt) 2, pp. 1434-7, 2004.

[7] Arroyo, E., Bonnani, L. & Selker, T. *Waterbot: exploring feedback and persuasive techniques at the sink.* Procedings of the Conference on Computer-Human Interaction (CHI), conference on Human Factors in Computing Systems, pp. 631-639, 2005.

[8] Toscos, T., Faber, A., An, S. & Gandhi, M.P. "Chick Clique: Persuasive Technology to Motivate Teenage Girls to Exercise". Extended Abstracts of International Conference on Human-Computer Interaction (CHI), 1873-8, 2006.

[9] Consolvo, S., McDonald, D.W., Toscos, T., Chen, M.Y., Froehlich, J., Harrison, B., Klasnja, P., LaMarca, A., LeGrand, L., Libby, R., Smith, I., & Landay, J.A. *Activity Sensing in the Wild: A Field Trial of UbiFit Garden*. Procedings of the Conference on Computer-Human Interaction (CHI), 2008.

[10] Niebuhr, S. & Kerkow, D. *Captivating patterns – A First Validation*. Revised Selected Papers of International Conference on Persuasive Technology, LNCS 4744, pp. 48-54, 2007.

[11] Out, J.W. *Baby Think It Over: Using role-play to prevent teen pregnancy, Adolescence*. 36(143): pp. 571-582, 2001.

[12] McCalley, T. & Mertens, A. *The Pet Plant: Developing an Inanimate Emotionally Interactive Tool for the Elderly*. Revised Selected Papers of International Conference on Persuasive Technology, LNCS 4744, pp. 68-79, 2007.

[13] Reitberger, W., Ploderer, B., Obermair, C. & Tscheligi, M. *The PerCues Framework and Its Application for Sustainable Mobility*. Proceedings of International Conference on Persuasive Technology, LNCS 4744, pp. 92-95, 2007.

[14] Oinas-Kukkonen, H. & Harjumaa, M. *A Systematic Framework for Designing and Evaluating Persuasive Systems*. Proceedings of International Conference on Persuasive Technology, LNCS 5033, pp. 164-176, 2008.

[15] Weiser, M. *The computer for the 21st century*. Scientific American, 265(3), pp. 94-104, 1991.

[16] Intille, S. S., Tapia, E. M., Rondoni, J., Beaudin, J., Kukla, C., Agarwal, S., Bao, L., & Larson, K. *Tools for Studying Behavior and Technology in Natural Settings*. UbiComp, LNCS 2864, pp. 157–174, 2003.

[17] Chi, P., Chen, J., Chu, H., & Lo, J. *Enabling calorie-aware cooking in a smart kitchen*. Proceeding of Persuasive, LNCS 5033, pp. 116-127, 2008.

[18] Bickmore, T., Mauer, D., Crespo, F., & Brown, T. *Persuasion, Task Interruption and Health Regimen Adherence*. Revised Selected Papers of International Conference on Persuasive Technology, LNCS 4744, pp. 1-11, 2007.

[19] Vogel, D. & Balakrishnan, R. *Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users*. Proceedings of the 17th annual ACM symposium on User interface software and technology, Santa Fe, NM, US, October 24-27, 2004.

[20] Streitz, N.A., Röcker, C., Prante, Th., Stenzel, R. & van Alphen, D. *Situated Interaction with Ambient Information: Facilitating Awareness and Communication in Ubiquitous Work Environments*. International Conference on Human-Computer Interaction (CHI), June 22-27, 2003.

[21] ACM SIGCHI curricula for human-computer interaction. New York, NY, USA, 1992.

[22] Dix, A., Finlay, J., Abowd, G. & Beale, R. *Human-Computer Interaction,* third edition**.** Prentice Hall. ISBN 0-13-239864-8, 2004.

[23] Preece, J., Rogers, Y. & Sharp, H. *Interaction Design: Beyond Human-Computer Interaction*. New York, NY: John Wiley & Sons, pp. 13-20, 2002.

[24] Boehm, B. *A spiral model of software development and enhancement*. SIGSOFT Softw. Eng. Notes, 11(4), pp. 14–24, 1986.

[25] Read, J. C., Markopoulos, P., Parés, N., Hourcade, J. P. & Antle, A. N. *Child computer interaction*. Extended Abstracts of International Conference on Human-Computer Interaction (CHI), Florence, Italy, 5-10 April, 2419-2422, 2008.

[26] Read, J. C. *Validating the fun toolkit: an instrument for measuring children's opinions of technology*. Cogn. Technol. Work, 10(2), pp.119–128, 2008.

[27] Debold, E. *Flow with soul: An interview with Dr. Mihaly Csikszentmihalyi*. What Is Enlightenment Magazine, pp. 108–118, 2002.

[28] Chen, J. *Flow in games (and everything else)*. Communications of the ACM, 50(4):31–34, 2007

Annex A - Smart Bins Creation

Description:

This test's objective is to set up the settings to a ubiquitous (being user interaction based on the use of sensors, without the use of usual computer inputs) application for kids to learn more about recycling in an entertaining way. The Smart Bins application is a game in which the objective is to place different recycling objects in the right containers to teach children the benefits of their actions towards the environment. This framework that we've developed for persuasive systems is generic but it requires the user to have an idea of how the application will work and also to have the multimedia contents for the menus created, so I will briefly explain how the game will work. To build this game, you have to fill form menus that correspond to the different game frames. The thing you'll have to pay more attention is the way changes occur between frames. It can be done in four different ways, using:

- Infrared sensors to change frame according to distance – to each one of the infrared sensors users may select the distance from it to change to the next menu.
- Camera to identify **Data Matrix** images - the camera is prepared to detect Data Matrix images that have to match the names defined by the user in a list of possible images to identify. If the name in Data Matrix image is part of the list, the frame of the game will change.
- Jump to other frame - to increase possibilities for interaction frames to change, users may choose to change frame after a specific period of time.
- Frame Appearances – the system will change frame after certain frame occurs a number of times defined by the application creator

As this game has a camera to identify **Data Matrix** images, you have to create lists of images to be identified. Each list corresponds to the images you want the camera to identify. In this game, we will have a user list and lists that correspond to recycling containers:

- Users – here, you'll have the elements the images that correspond to user's names (ex: Manuel, Madalena, Pedro)
- Green Container – the only element on this list will be glass for now
- Blue Container – the only element on this list will be paper for this test
- Yellow Container - the elements on this list will be plastic and metal

Note: These lists will correspond to images that users have to print to play the game

To better understand the game you have a description of the frames that may occur during user interaction.

1 – First you have to create a welcome frame. This frame is a stand-by state before the actual game which will start when people approach the system. In this frame there's a video about recycling playing until someone gets near the system. If someone approaches the sensor the system changes to the next menu.

2 – For this frame, there's another background image with a text description of the game and an audio explanation. In this menu this application waits until someone shows an identification card to the camera. (Remember – you have to add a name for the list and the names of users that will use this game).

3 – When the camera identifies the user's image, the background image changes again. This frame also contains more information about the game and a text message that welcomes the new user. Users are now asked to identify the first recycling object to the camera to begin the game. The system will change to a different menu if the identified image corresponds to glass, paper, plastic or metal.

4, 5, 6 - In this frames the system has to identify which container the identified object is going to be placed in. Each container: Green, Blue or Yellow has a hidden sensor placed in its interior. You have to know that IR Sensor 2 is at the Green container, IR Sensor 3 at Blue container and IR Sensor 4 at the Yellow container. An object is detected by one of these sensors when the object is thrown inside the container.

7 – This is the frame that appears to congratulate the users that have placed the object in the correct container. There's also and applause sound in this menu. After 5 seconds on this frame, the game changes to frame 9.

8 – If the user doesn't place the object correctly this is the frame shown. Like before, after 5 seconds the system advances to frame 9.

9 – This frame has the same objective as frame 3 (changes to frame 4, 5 or 6 depending on the identified material type), but as the background image is different, it has also to be defined.

10 – That's it. This is the final frame. The score will be presented to the user as some more useful information about recycling.

Smart Bins Development

Run the application. Choose Create / Edit Persuasive Interfaces

Create an application with these settings:

Theme: Smart Bins	Infrared Sensors - Yes
Number of Frames: 10	Data Matrix - Yes
Window Resolution: 1280x770	Timer - Yes Time - 300s

When you are done with these settings you have to create the menus for the game. Given the storyboard available in the next page, you have to create the settings for the interaction menus that hopefully will result in the desired game. Each point represents a menu for the game and all the settings necessary for it are presented in the storyboard in the next pages.

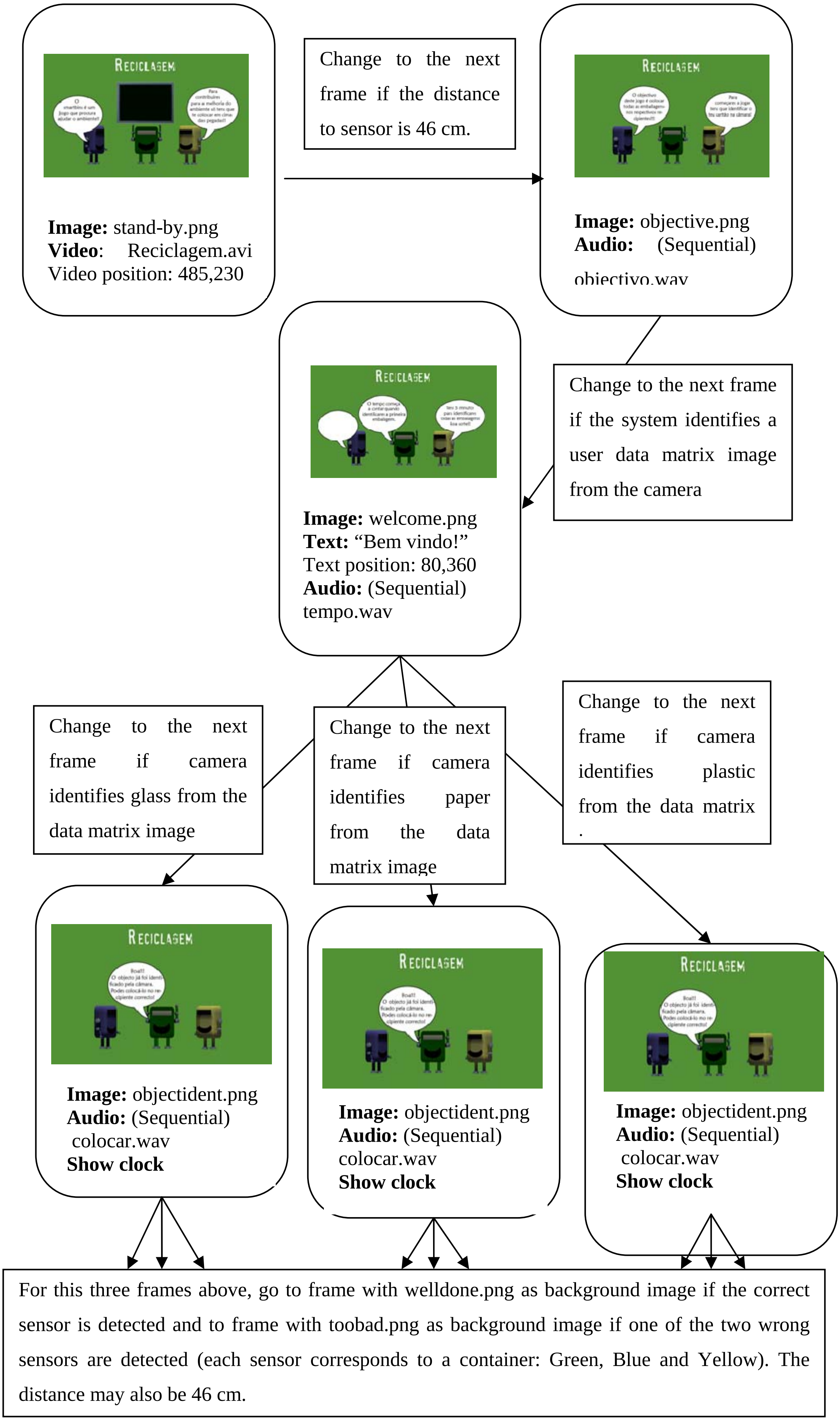
Reciclagem
Image: stand-by.png
Video: Reciclagem.avi
Video position: 485,230
Change to the next frame if the distance to sensor is 46 cm.
Reciclagem
Image: objective.png
Audio: (Sequential)
objectivo.wav
Change to the next frame if the system identifies a user data matrix image from the camera
Reciclagem
Image: welcome.png
Text: "Bem vindo!"
Text position: 80,360
Audio: (Sequential)
tempo.wav
Change to the next frame if camera identifies glass from the data matrix image
Change to the next frame if camera identifies paper from the data matrix image
Change to the next frame if camera identifies plastic from the data matrix
.
Reciclagem
Image: objectident.png
Audio: (Sequential)
colocar.wav
Show clock
Reciclagem
Image: objectident.png
Audio: (Sequential)
colocar.wav
Show clock
Reciclagem
Image: objectident.png
Audio: (Sequential)
colocar.wav
Show clock
For this three frames above, go to frame with welldone.png as background image if the correct sensor is detected and to frame with toobad.png as background image if one of the two wrong sensors are detected (each sensor corresponds to a container: Green, Blue and Yellow). The distance may also be 46 cm.

Image: welldone.png
Text: “Boa!”
Position: 80,360
Audio: (Sequential)
aplauso.mp3
Show clock

Image: toobad.png
Text:“Oh!”
TextPosition:80,360
Audio:(Random)
oh.mp3
oh1.mp3
Show clock

Change to the new frame after 5 seconds

Change to next frame after 5 seconds

Image: nextobject.png
Show clock

Change to the final frame after being 7 times in the frame above

This frame has the same links as in the menu with bg3 to create a loop effect

Image:
congrats.png

Jump to first frame after 40 seconds