OPEN Project

STREP Project FP7-ICT-2007-1 N.216552

Title of Document:	Initial application requirements and design
Editor(s):	Stefano Bigi, Giancarlo Cherchi, Dario Deledda, Francesca Mureddu
Affiliation(s):	Arcadia Design
Contributor(s):	Agnese Grasselli, Holger Klus, Stefano Marzorati, Carmen Santoro
Affiliation(s):	Clausthal, CNR, Vodafone
Date of Document:	October, 2008
OPEN Document:	D5.1
Distribution:	EU
Keyword List:	Applications, prototypes, scenarios, design, requirements
Version:	2.3

OPEN Partners:

CNR-ISTI (Italy)
Aalborg University (Denmark)
Arcadia Design (Italy)
NEC (United Kingdom)
SAP AG (Germany)
Vodafone Omnitel NV (Italy)
Clausthal University (Germany)

Title: Initial application requirements and design	**Id Number: WP5 – D5.1**

Abstract

This deliverable describes the guidelines for defining the prototype applications to be developed during the course of the project. This is an ever-evolving reference document that will be finalized in D5.3 - Final application requirements and design, and it should be considered as a tool for communicating the vision of the applications to the consortium, in order to share among all the partners what the application prototypes are supposed to be.

The document is organized into three sections, the first one taking into consideration the general application point of view in the project, the second one and the third one detailing the two considered application domains, namely the business and game domain, respectively. In the current version, only the game domain is described, whereas the business domain will be covered in another release of this deliverable.

Title: Initial application requirements and design	**Id Number: WP5 – D5**

Table of Contents

1. INTRODUCTION 2

2. GOALS AND MOTIVATIONS 3

3. BUSINESS DOMAIN 4

4. GAME DOMAIN 5

4.1. PRELIMINARY ANALYSIS 5
4.1.1. Arcade Game 5
4.1.2. Social Game 7
4.1.3. Technical Experiments 7
4.2. CASE STUDY 8
4.2.1. Scenario 8
4.2.1.1. Situation 8
4.2.1.2. Storyboard 9
4.2.2. Migration aspects 13
4.2.2.1. From college library to home 14
4.2.2.2. From the living room to the kitchen 16
4.2.2.3. Change of user 17
4.2.2.4. At the pub 18
4.2.3. Requirements 19
4.2.4. Early design 27
4.2.4.1. IPTV 29
4.2.4.2. Chat 29
4.2.4.3. Betting 29
4.2.4.4. Additional content 29
4.2.4.5. Game 30
4.2.4.6. Other functionalities 31
4.3. FIRST YEAR PROTOTYPE 31
4.3.1. Arcade Game 32
4.3.2. Social Game 36

5. CONCLUSIONS 38

6. REFERENCES 40

1. Introduction

The aim of this deliverable is to describe and discuss the guidelines for defining the prototype applications that will be developed during the course of the project. It can be considered as a tool for communicating the vision of the applications to the consortium, in order to share among all the partners what the application prototypes are supposed to be. In order to cover a wide range of migration aspects, two different domains, which shows complementary characteristics and attributes have been analyzed, the business domain and the game domain.

Since this is an ever-evolving reference document that will be finalized in D5.3 - Final application requirements and design, this first version mainly considers the applications that will be released in the next deliverable, D5.2 - Initial prototype applications.

We will not give in-deep technical specifications, rather we will mainly focus on scenarios descriptions, requirements elicitation, and high level design, to set guidelines and ideas for further developments. The design itself can be subject to change during the entire project development process.

The document is organized into three sections. The first one considers the general application point of view in the project and describes the main goals and motivations. The second one and the third one are dedicated to detailing the two considered application domains, namely the business and game domain, respectively.

It is worth noting that, in the current version, only the game domain is described, whereas the business domain will be covered in another release of this deliverable.

2. Goals and Motivations

The requirements and the design of the two application domains are the consequence of the work in WP1 - Requirements and Scenarios.

The starting point has been, from one hand, the analysis of the several scenarios contained in D1.1; from the other hand, the architecture description in D1.2.

The scenarios have covered both the domains and the requirements elicited from them, whereas the architecture description has driven the technological considerations and the main technical requirements.

WP5 has the goal of experimenting the applicability of migratory interactive services with application prototypes in two different application domains, namely business applications and games. The task is a difficult one, since both the migration-enabling technology needed for the development and existing application experiences are not available. In fact, the migration-enabling technology will be developed during the course of the project. Regarding the concept of migration in applications, for the game domain it is totally new, and no concrete examples of such application experiences are available to date.

The approach followed for the design and the development of the prototypes is an iterative one, with two main releases.

The goal of the initial prototypes is to assess possible development impacts and roadmaps, both from the migration and the application experience points of view, starting from an analysis in practice of the theoretical scenarios suggested in WP1. In particular, they are useful to identify which kinds of applications are most suitable for migration, both for the final user and the application developer. Moreover, the initial prototypes will help in better understanding how the applications should be adapted to support the migration platform technology. Finally, the prototypes could highlight the possible issues that are likely to be faced during the integration with the migration platform.

Since the development of the initial application prototypes runs in parallel with the development of the first migration platform related prototypes, any form of integration with the migration platform will hardly be achievable. For this reason, the initial prototypes will be mainly based on simulated technology. This preliminary version will help to establish a comparison between the needed technology for the desired migrable application and the capabilities of the platform.

Of course, the design of the initial prototypes will also take into account the full “big picture” to be accomplished at the end of the project, both regarding the user experience and the technical development, iterating back and forth across the intermediate releases. The first iteration is being conducted by developing a simple incremental prototype to test the functionalities developed in WP2, WP3 and WP4, and to provide feedback to these WPs for the subsequent iterations.

3. Business Domain

4. Game Domain

As mentioned in section 2, games that support migration among different devices are not available to date. Starting from the scenarios analysed and proposed in D1.1, together with the corresponding migration cases, two different types of game experiences, which represent complementary set of requirements, have been considered. Besides analysing possible game experiences, some technical investigations on the feasibility of specific application components have been also made.

4.1. Preliminary Analysis

To better investigate the possible implications of the considered scenarios towards the requirement elicitation process, a preliminary analysis involving different kind of gaming experiences have been made. The following sections describe first a classic and well-known game, then a kind of modern game experience, and finally some possible technical approaches that could be related to migration functionalities.

4.1.1. Arcade Game

Arcade games are well known games that have proven to be extremely popular among gamers. In fact these games often have very short levels, simple and intuitive control schemes, and rapidly increasing difficulty. Since the game experience is well known, in principle to conceive some fruitful and interesting migration case is quite straightforward. At the same time, arcade games are relatively simple applications, so that also the management of the application is not extremely difficult. In this way, by leaving aside possible game issues, the focus has been kept on some migration aspects, which have been isolated and separately examined. The experience made with arcade games will then help to better understand potential issues on the migration, as well as adapting the game itself to migration; moreover, it will allow starting improving the available migration technology for supporting migration of games. At first a game similar to the popular PAC-MAN has been considered to investigate UI migration and logic configuration aspects.

Regarding UI migration aspects, since the current available technology was only able to migrate web pages, as a starting point for carrying out the common exercise, a web version of the PAC-MAN implemented in HTML and JavaScript has been identified and further adapted to the current migratory technology. Figure 4.1 shows a screenshot of the game.

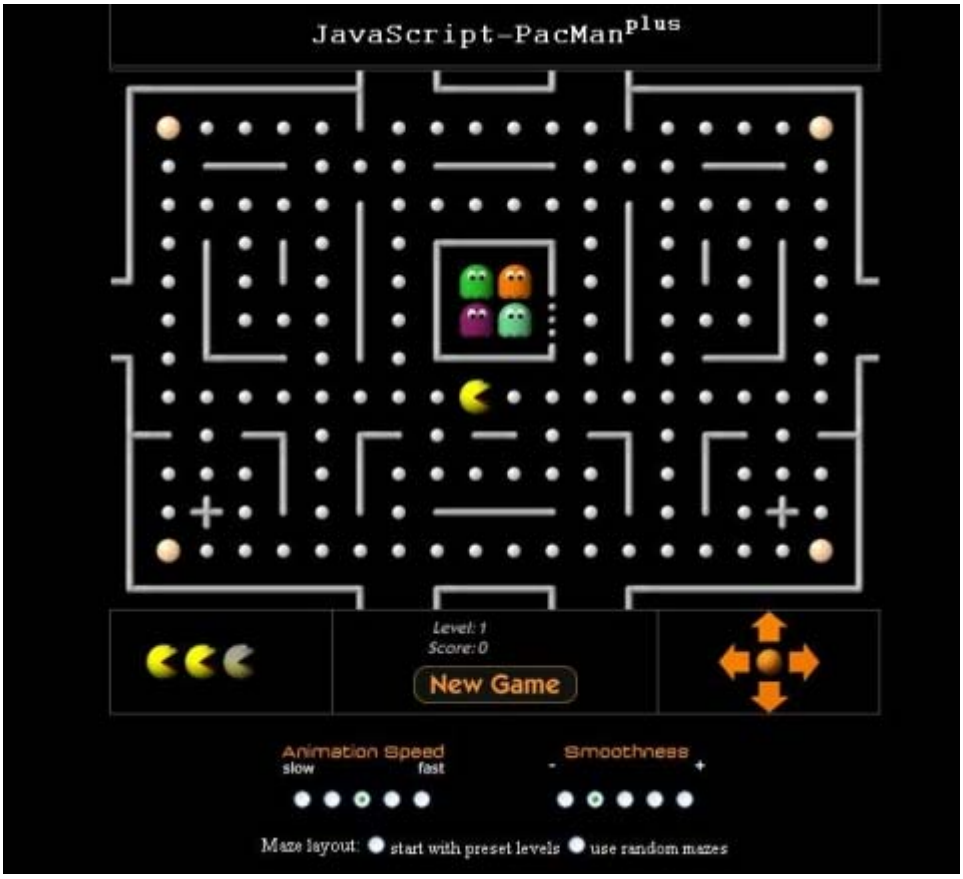

Figure 4.1: The pacman game

The considered game uses HTML language basically for rendering the user interface, and JavaScript code for supporting the application logic (e.g. defining the behaviour of the pacman, ghosts, etc.) and also for enabling additional user interaction options (e.g. setting the animation speed of the pacman).

During a first analysis, some issues have been raised, in particular regarding the mark-up validity of the web page considered, and the fact that some XHTML constructs were not appropriately used in that page (e.g. some elements were used only for conveying particular formatting effects). Such issues could bring accessibility problems as well as preventing from conveying the intended structure when the page has to be rendered to different devices. In addition, in the original version of the game, some UI elements, such as table row, table columns and imagemaps, were modelled in a way that was not deemed appropriate for the experiment: for instance, in the original version of the page, multiple images were contained in a single table cell.

A first iteration has been done with an improved version of the page, where the abovementioned issues have been solved, for example only one element has been assigned to each table cell. After this iteration, the web page of the pacman game for the desktop platform has to be transformed in order to obtain a new version adapted for the mobile device. To do this, a reverse engineering of the pacman web page for the desktop platform, in order to obtain a more abstract UI description of the page (the so-called concrete UI), is needed. Such a description

will then be redesigned to finally generate the adapted user interface for the mobile device.

However, the analysis of the output produced by the reverse engineering module applied to the pacman page, has led to the identification of some XHTML elements that were not yet supported by reverse engineering module itself (e.g.: imagemaps). To this aim, new constructs have been included in an updated version of the language describing the concrete user interface, and new methods have been included in the reverse engineering module algorithm. However, as it is an ongoing work, it is likely that further indications and guidelines will be identified while progressing the exercise carried out on the considered pacman game, especially if further updated versions of the game will be analysed.

Regarding logic configuration, during the migration of an application, the application itself has also to be adapted to the target device, e.g. to available resources like CPU rate or storage capacity. Therefore, the PacMan scenario has also been used to identify possible adaptation techniques, in particular for strategies and possibilities to adapt the application logic in order to provide a better user experience. To this end, an existing middleware for application logic reconfiguration has been applied to a Java-based PAC-MAN game implementation. The gathered experiences will be used to improve techniques for application logic reconfiguration during migration, e.g. by considering context information during adaptation as well.

4.1.2. Social Game

Opposite to a well known game, it has been chosen to investigate a more modern and innovative game experience, some that could possibly represent a game of the future. In this context, it is very difficult and definitely not clear a priori how to conceive migration scenarios. At the same time, migration services may represent a meaningful component of the overall game experience.

For these reasons, a “temporarily” persistent game complemented with social networking and a series of elements coming from different entertainment domains have been selected. As users form a community that play and interact for a prolonged period of time by using different media and application functionalities, migratory services enabling a continuous access across different terminals and conditions may play a fundamental role in the overall experience with the “entertaining” application. The IPTV scenario discussed in WP1 and described in D1.1 (section 7.1.5) has been appropriately modified and adapted to obtain a more general case. This new scenario will be discussed in the next section. This line of work represents the “grand vision” of a migratory game and it should be considered as the main tool to study possible migration cases, aimed at suggesting ideas for the design and the development of incremental prototypes.

4.1.3. Technical Experiments

In order to develop migration functionalities, an important part of the work is to assess possible issues and implications in the development process. Since games application can be very demanding in terms of computational resources, one of the first possible migration cases that has been analysed from a technical point of

view was the migration from a single-core device to a multi-core one. The main idea was to exploit a multi-core framework integrated within the game engine on which the game would be based. Several tests have been performed aimed at identifying which components would have benefited from the exploitation of such framework capabilities. The tests showed that the proposed ideas would not match with OPEN topics and research goal.

4.2. Case Study

The following paragraph describes the “grand vision” of a migratory game and it should be considered as the main tool to study possible migration cases, aimed at suggesting ideas for the design and the development of incremental prototypes. It constitutes the main case study for extracting a set of requirements and sketching a preliminary design of the prototypes. It is important to remark that it is not meant to be a real prototype implementation reference. Rather, this scenario will serve as the basis for making considerations about the design and the development of the possible simplified prototypes that may be realised within the project.

4.2.1. Scenario

Thomas is a 19 years old college student who loves Formula 1. He is a big fan of Lewis Hamilton, and spends 4/5 hours a day playing video games. He is used to watching all F1 Grand Prix races on his laptop with LiveF1 software: he usually places it on the couch and watches the race while playing video games on his game console connected to the Plasma TV. ACME Corporation has just announced the release of the Connected Entertainment package that merges its IPTV services with the game console. Thomas purchases the Connected Entertainment package, as the F1 new season is only two weeks away, and he upgrades his Formula 1 Championship Edition game license with all the new season official drivers, cars, circuits, rules and teams.

4.2.1.1. Situation

As illustrated in Figure 4.2, Thomas has left the study room of his college library just a few minutes before the start of the first Grand Prix of the season. He starts playing with the mobile phone while waiting for the bus on his way home. Thomas checks Brad’s presence status in order to verify if he is already playing. Brad is online but he is not playing, so Thomas sends him an instant message (IM) to invite him to be ready for the race. As Thomas gets home, the Set-Top Box (STB) sends a message to the mobile to suggest a migration. Thomas accepts and the context is switched to the IPTV-enabled gaming console (STB). Since he has the gaming and chatting sessions activated, the same contents are displayed in his Plasma TV. The STB receives game commands from the phone/remote control, sends the commands to a server in the network and receives the images from a streaming server on the network. The Grand Prix is going to start, so Thomas opens another HD window on the Plasma TV, to watch the Grand Prix. Everything is ready! Thomas loves it. With his IPTV-enabled gaming console, he can watch high-definition F1 Grand Prix on one side of the screen, and on the other side, he

can virtually race his own car against the real contestants (Raikkonen, Massa, Alonso, but above all Hamilton).

He invites Brad, who is still at the library. Waiting for Brad to join the game, Thomas opens the internet contents to have a look at the weather forecast for the race. Then he decides to bet: he checks the tipsters and bet on Hamilton as winner using his prepaid amount on the mobile phone. The Grand Prix starts (see Figure 4.3), so in the social networking window he swaps from betting to the chat and he collapses the chat. Brad plays the game on his mobile phone without watching the Grand Prix, Thomas plays while watching it. Suddenly his grandfather enters the sitting room, asking for help: Thomas should hold the ladder while his grandfather tries to fix a broken ceiling lamp in the kitchen. Thomas has to make an unscheduled pit stop, because he cannot go on playing while holding the ladder; he decides to migrate the IPTV to the kitchen LCD and the game to the phone screen. Since he has done a pit stop, automatically only the game dashboard is migrated to the phone screen to keep Thomas informed of his car's state (lap, position, etc.).

As the lamp fixing is lasting longer than expected, Thomas decides to give up playing and quits the game from his mobile while still holding the ladder. In the meanwhile Brad, who is still playing on his mobile, after having left the college library, gets off the bus and, instead of going home, rings Thomas' door bell: Thomas' grandmother opens the door and Brad enters in the living room. Brad sees that Thomas left open his chat and asks him if he can connect to the STB. Thomas from his mobile phone closes his chat session and since Brad's mobile is a registered device in the STB, Brad's game seamlessly switches from his mobile to the STB. After twenty minutes Thomas gets back, just in time to see Hamilton winning the Grand Prix on one side of the screen and Brad coming in second on the other side of the screen (see Figure 4.4).

A scenario variant can be occurring in a pub, where players are watching the real race on a big screen. When a user wants to play the game, he asks to the STB to join. After that the STB verifies if the user can join, the screen is split in several parts, each of them being dedicated to one of the players attending the game, and showing a real-time simulation generated by a centralized server. Each player can control the game with her/his mobile phone. The private contents, such as chatting and betting, are displayed on the personal mobile phone (see Figure 4.4).

4.2.1.2. Storyboard

One of the best ways to design a game is to *storyboard* it, by creating a sequence of drawings that show the levels of the game and the different scenes and goals. In order to better understand the game scenario implications, the same process has been applied to the considered scenario, through a storyboard that describes the various situations, with special attention to involved actors, devices, migration aspects and the other application specific elements.

In particular, four situations have been identified, each of them involving different types of migrations that will be analysed in more detail in 4.2.2: (1) From college library to home, (2) From the living room to the kitchen, (3) Change of user, (4) At the pub.

Figure 4.2: Scenario (part 1)

Figure 4.3: Scenario (part 2)

Figure 4.4: Scenario (part 3)

4.2.2. Migration aspects

In this paragraph, the migration aspects regarding the application derived from the scenario will be presented. The scenario has been analysed with the help of the storyboard, in order to extract involved actors, devices and application components. Starting from the elements of the application, the components involved in the migration have been identified, with special attention to those related to the user interaction. The aim of this analysis is not to provide technical details about how to implement migration functionalities, but only to give a case study for further deepening about migration related issues.

Actors	**Devices**	**Elements**
Thomas	Thomas' mobile phone	Game
Brad	Brad's mobile phone	Chat
	STB + Plasma TV	IPTV
	LCD Screen	Betting
	STB + Big Screen	Additional Content

Table 4.1: Actors, devices, and elements in the IPTV scenario

Table 4.1 shows the actors, the devices and the application elements involved in the IPTV scenario.

The application has been split into different elements, each of them being separated in turn in several components that can be separately migrated.

Figure 4.5 shows the main elements of the application with the corresponding components.

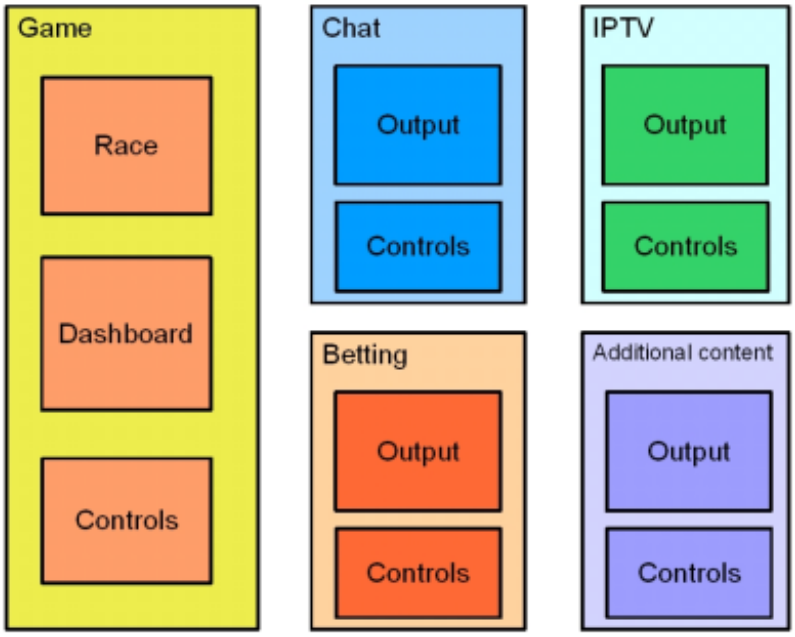

Figure 4.5: Main elements and components of the application

The Game can be considered as composed of three elements: the Race, which represents the real time game, the Dashboard, by which the player can login, select the car and so on, and the Controls, with which the player interacts with the game itself.

All the other application elements have two components: the Output and the Controls.

In the following, the specific migration cases involved in the IPTV scenario will be described accordingly to the classification suggested in D1.2, in which various migration categories have been described, in particular for User Interface and Application Logic Adaptation (section 1.2) and Migration Types (section 2.1). It is worth noting that some of the presented migration cases are very complex, and it is not easy to embrace them in a single described migration category. Nevertheless, they can be considered as a combination of several categories. Moreover, since the final prototype architecture is still to be defined, in the presented cases, only the devices directly involved in the migration will be shown.

4.2.2.1. From college library to home

The first migration happens when Thomas arrives at home, while he is using the Chat and the Dashboard of the game in his mobile phone, as shown in Figure 4.6.

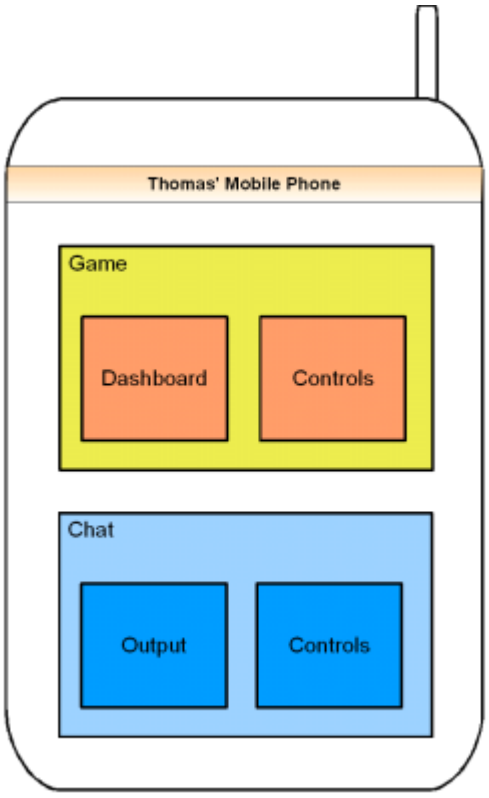

Figure 4.6: Components migrated to mobile phone

Once he arrives at home, the STB sends a message to his mobile phone to suggest a migration. The Game and the Chat migrate from the mobile phone to the STB, whereas the Controls stay on the mobile phone. After the migration, more functionalities become available on the STB: IPTV, Betting and Additional Content. The Controls of these new functionalities migrate from the STB to the mobile phone. Figure 4.7 shows the situation after the migration.

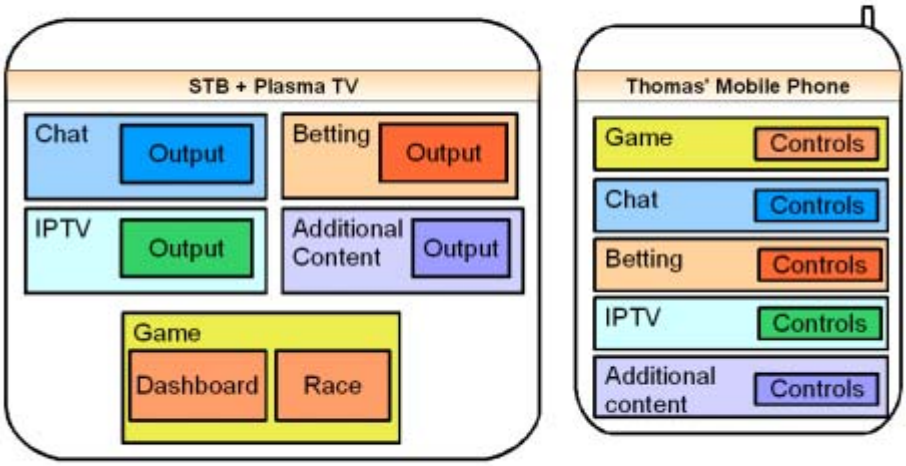

Figure 4.7: Situation after first migration

In this first migration, the following migration types can be identified:

- **Partial**: part of the Game and the Chat migrate from the Mobile Phone to the STB
- **Distributing**: IPTV, Betting and Additional content are distributed between the Mobile Phone (Controls) and the STB (Output)

From the UI point of view, the following different strategies may be considered:

- **Rearrangement, Increase** or **Magnification,** for the Chat
- **Magnification** or **Increase,** for the Game Dashboard

From the Application Logic point of view, there is an increase of functionality of the Game, which can be fully played only on the STB.

Regarding the context management, the migration is suggested by one of the devices and triggered by the user.

4.2.2.2. From the living room to the kitchen

The second migration happens when Thomas moves from the living room to the kitchen. The situation before the migration is shown in Figure 4.7: the IPTV migrates to a LCD display in the kitchen, whereas the game Dashboard and the Chat migrate to his mobile phone, as shown in Figure 4.8.

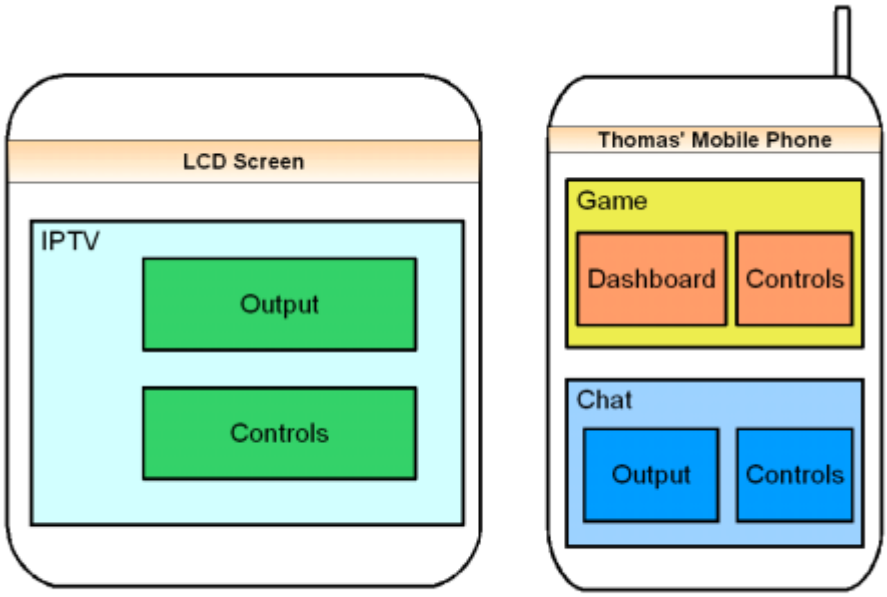

Figure 4.8: Situation after the migration

After the migration, only some of the functionalities are available. The following migration types have been identified:

- **Aggregating:** the IPTV migrates from Mobile Phone + STB to LCD display; the Chat and the Game migrate from Mobile Phone + STB to Mobile phone
- **Partial**: after the migration, only the Dashboard of the Game is available on the Mobile Phone.

From the UI point of view, the following strategies can be taken into consideration:

- **Conserving,** for the IPTV
- **Rearrangement**, **Reduction** or **Simplification,** for the Chat
- **Reduction** or **Simplification,** for the Game

As for the context management, in this case, the user decides the migration according to the available devices.

As for the Application Logic, a decrease of functionalities is considered for the Game, since in the Mobile Phone only the Dashboard functionalities are available.

4.2.2.3. Change of user

The third migration happens when Brad arrives at Thomas' home and asks Thomas to use his STB to play. Thomas disconnects from the game and Brad migrates his game from the mobile phone to the STB, while the IPTV migrates from the LCD display in the kitchen to the Plasma TV in the living room. This

migration creates a situation similar to that of Figure 4.8, but in this case there are a change of user (Brad instead of Thomas) and a migration of the IPTV from the LCD display to the Plasma TV.

Beyond the migration types described beforehand, the **Conserving** type has to be considered for the IPTV.

Moreover, the Context Management and Application Logic Reconfiguration are highly involved in the change of the user.

4.2.2.4. At the pub

The migration happens when the players ask by their Mobile Phone the STB to join. In Figure 4.9, the situation before the migration is shown.

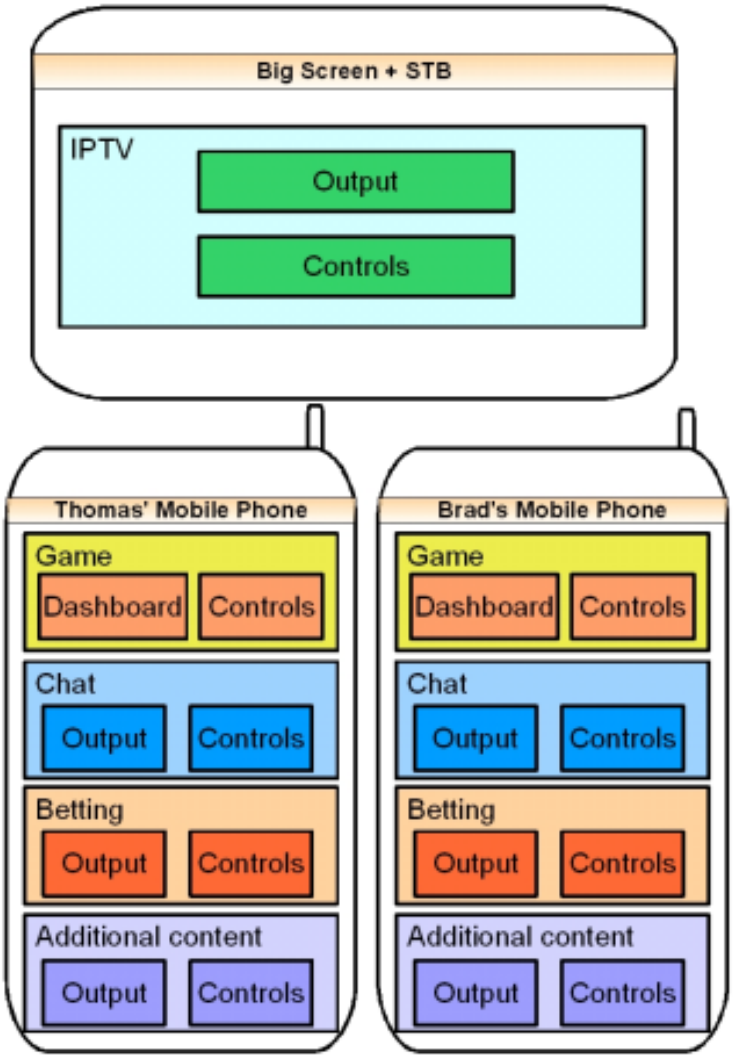

Figure 4.9: Situation before migration at the pub

Part of the Game is migrated from the mobile phone to the STB and the output of the race is shown on the Big Screen, while the private content such as Chat and Betting stays on the mobile phone (see Figure 4.10).

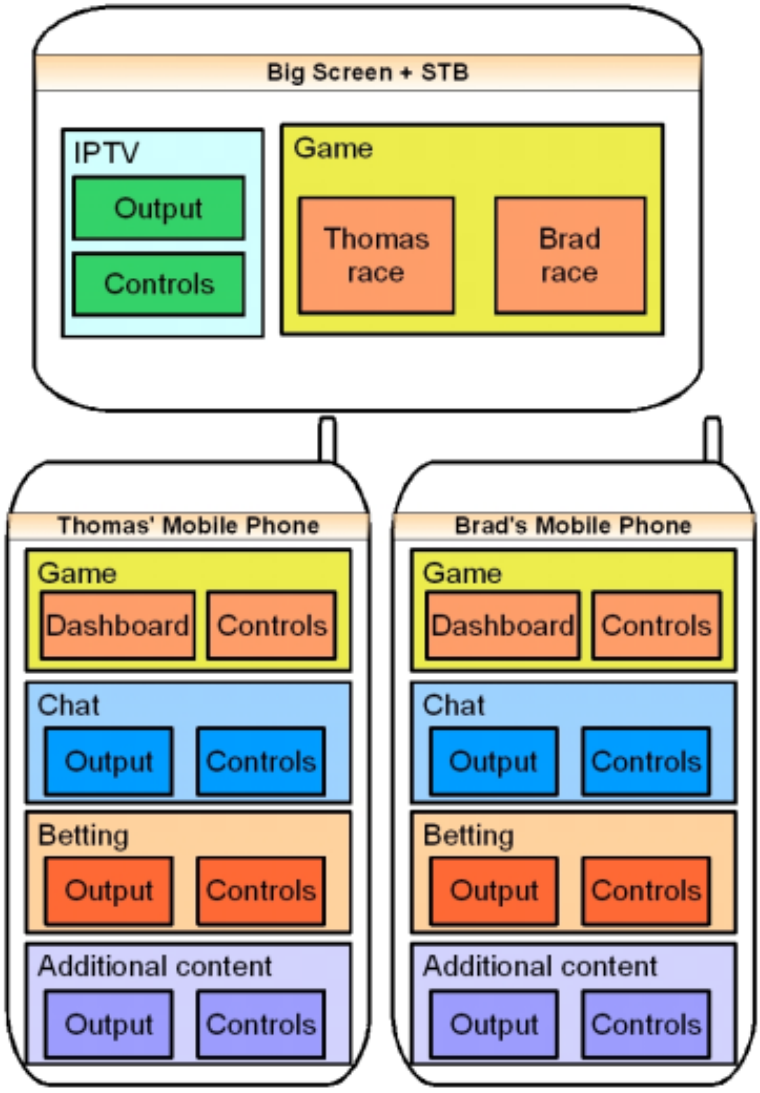

Figure 4.10: Situation after migration at the pub

The users watch the game and the live race in the Big Screen at the pub, and they control the game by their mobile phones.

As far as migration types are concerned, the following can be considered:

- **Partial:** part of the Game is migrated from the Mobile Phone to the Big Screen
- **Aggregating**: the Game migrates from the two Mobile Phones to the Big Screen.

Regarding the UI, there is an **Increase** on the Game side.

4.2.3. Requirements

The application requirements defined in this document are described adopting a variation on Volere Requirements Resources [VRR]. The Volere Requirements Specification Template is freely downloadable in a pdf version from the Volere

site [1] and can be adapted to a specific requirements gathering process and requirements tool.

The Volere Requirements are classified with the following categories:

- Functional
- Look and Feel
- Usability and Humanity
- Performance
- Operational
- Maintainability and Support
- Security
- Cultural and Political
- Legal

In this document the categories considered are:

- Functional: to specify the detailed functional requirements for the activity of the product, this can be either platform related or application related.
- Look and Feel: to capture the requirements for the appearance
- Operational: to specify requirements for testing and monitoring.

The following table gives a short explanation of the information provided.

ID	Requirement ID
Description	Short description of the current requirement.
Status	Status of this requirement report (e.g.: "Proposed"), Version and Phase numbers
Revision	Information about revision history *(e.g.: Created on* 16/6/2008. *Last modified on 4/7/2008)*.
type	One of the requirement categories (Functional, Look and Feel, Operational)
Rationale	The rational behind this requirement. (OPTIONAL**)**
Priority	Level of priority about the fulfilment of this requirement (Low, Medium, High).
Author category	Telco, SW development, Game player…

ID	1
Description	If the application is running on the mobile Phone, then: • IPTV is not available in the mobile screen and it is eliminated from the application's options • Several tabs enable the user to switch to social betting, messaging, gaming or internet access options from the mobile phone.
Status	Proposed
Revision	*Created on* 08/09/2008. *Last modified on 08/09/2008*
Requirement type	Functional
Rationale	There can be usability issues and network resources constraints for having IPTV service on mobile
Priority	Medium
Author category	Telco

ID	2
Description	The migration can be: • Platform initiated: o the platform proposes the destination device and the UI components to migrate o the user can confirm or reject the migration o in case the user rejects the migration, she/he can eventually proceed with a user initiated migration • user initiated towards the OPEN platform: o the OPEN platform automatically proposes a list of available devices and the UI components o the user selects the destination devices and the UI components to migrate
Status	Proposed
Revision	*Created on* 08/09/2008. *Last modified on 08/09/2008*
Requirement type	Functional
Rationale	N/A
Priority	High
Author category	Telco

ID	3
Description	The migration can be: • Total, all components of the application (i.e. betting, messaging, gaming or internet access option) migrate • Partial, only a part of the application migrates (i.e. the dashboard of the game application or the betting option)
Status	Proposed
Revision	*Created on* 08/09/2008. *Last modified on 08/09/2008*
Requirement type	Functional
Rationale	N/A
Priority	High
Author category	Telco

ID	4
Description	The visualization for large screen can be: • Single user • Multiple user
Status	Proposed
Revision	*Created on* 08/09/2008. *Last modified on 08/09/2008*
Requirement type	Functional
Rationale	N/A
Priority	High
Author category	Telco

ID	5
Description	For Single user visualization: • The tabs displayed are: o Game o Messaging (chat, betting with tab to switch and chat as the default) o Internet access o IPTV o Display settings

	▪ Picture in Picture (PIP) between IPTV and game ▪ Fixed schema ▪ For PIP switch function between IPTV and gaming o Migration, with the previous requirements • If no migration has been performed before, then the default visualization is IPTV • After a migration towards the large screen all previous selected services are activated
Status	Proposed
Revision	*Created on* 08/09/2008. *Last modified on 08/09/2008*
Requirement type	Look and Feel
Rationale	N/A
Priority	High
Author category	Telco

ID	6
Description	For Multiple user visualization: • If no migration has been performed before, then the default visualization is IPTV • When other player join, the screen is partitioned according to the total number, with a maximum supported • Private contents, such as chat and betting, automatically migrate to mobile phone (private phone)
Status	Proposed
Revision	*Created on* 08/09/2008. *Last modified on 08/09/2008*
Requirement type	Look and Feel
Rationale	N/A
Priority	High
Author category	Telco

ID	7

Description	For Multiple user visualization: • There should be a way to define which the private device is
Status	Proposed
Revision	*Created on* 08/09/2008. *Last modified on 08/09/2008*
Requirement type	Functional
Rationale	N/A
Priority	High
Author category	Telco

ID	8
Description	As for betting, the user should be able to: • See tipsters and betting parameters • Check bet results • Pay by credit card or a mobile operator prepaid wallet
Status	Proposed
Revision	*Created on* 08/09/2008. *Last modified on 08/09/2008*
Requirement type	Functional
Rationale	N/A
Priority	Medium
Author category	Telco

ID	9
Description	The game can be played by a single or multiple users. The game dashboard can be visualized through widgets. Some migratory rules exist, being the following an example: when the user has done a pit stop, only the game dashboard is migrated automatically. One or more of the game opponents are represented by avatars, with positions determined by real telemetry data.
Status	Proposed
Revision	*Created on* 08/09/2008. *Last modified on 08/09/2008*

Requirement type	Functional
Rationale	N/A
Priority	High
Author category	Telco

ID	10
Description	For both large and small displays, the messaging features allow the user to: • See the buddies list (with presence information such as: on-line, off-line, gaming, watching TV with channel info) • Start chat sessions • chat
Status	Proposed
Revision	*Created on* 08/09/2008. *Last modified on 08/09/2008*
Requirement type	Functional
Rationale	N/A
Priority	Medium
Author category	Telco

ID	11
Description	IPTV capabilities are: • streaming • EPG (Electronic Program Guide) with channel selection • additional contents linked to TV channels • change view (switching cameras during race streaming)
Status	Proposed
Revision	*Created on* 08/09/2008. *Last modified on 08/09/2008*
Requirement type	Functional
Rationale	N/A
Priority	Medium
Author category	Telco

ID	12
Description	Every part that cannot be migrated to the new device because of hardware constrains, could be migrated in an alternative way. For instance, when Brad is helping his father and carrying the mobile phone with him, he could receive a vibration for every laps that he is missing,
Status	Proposed
Revision	*Created on* 29/9/2008. *Last modified on 29/9/2008*
Requirement type	Functional
Rationale	N/A
Priority	High
Author category	SW development

ID	13
Description	During the migration process, the application should have a stand by or auto-drive option that allows the user to have no disadvantage for migrating the application.
Status	Proposed
Revision	*Created on* 29/9/2008. *Last modified on 29/9/2008*.
Requirement type	Functional
Rationale	N/A
Priority	Medium
Author category	User

ID	14
Description	When the game is migrated in a device with no resources for the rendering, the game is rendered in a server in the network and the rendering will be streamed towards the device.
Status	Proposed
Revision	*Created on* 29/9/2008. *Last modified on 29/9/2008*.
Requirement type	Functional

Rationale	N/A
Priority	Low
Author category	Telco

ID	15
Description	Create a Log file with the purpose of measuring the framework and application execution (see D6.3).
Status	Proposed
Revision	*Created on* 29/9/2008. *Last modified on 29/9/2008*.
Requirement type	Operational
Rationale	N/A
Priority	High
Author category	Telco

4.2.4. Early design

According to the described scenario and the proposed requirements, the desired application is expected to be a very complex one. In fact, it should be a rich web application embedding several different elements: a complex 3D game, an IPTV, a set of advanced social networking functionalities. Targets of the scenario are racing sports fans and players of racing games in general. A user who is usually a spectator discovers a more interactive way of watching and enjoying the live race. The typical game player improves his experience by challenging the professional pilots in a real context.

Of course, it would be extremely complex to cope at the same time with all the aspects involved in the entire scenario and the related migration issues, as this is outside the objective of the project. Our goal is to provide a testbed for migratory technologies instead of developing a complete application, and to this end the early design is toward. Hence, only a subset of the desired requirements will be considered, and the application's functionalities will be simplified or simulated in the prototypes.

In other words, the design process will mainly focus on the functionalities that are more relevant with respect to the migration issues, nevertheless taking into account the big picture, although the prototypes might disregard some specific features on a case by case basis.

To restrict the possibilities and making the general design more practicable, some initial assumptions have to be made. In particular, the first step is to decide which kind of application technology should be adopted for the prototypes development. For this reason, among the possible choices regarding the technologies on which the application prototypes have to be built for testing scenario's requirements, a

web based application has been selected. Note that this choice is not restraining, since web-based applications are becoming more and more common and, in principle, any stand-alone application can be adapted to a web container, with some minor restrictions mostly on the user interface. Moreover, using a web-based approach, several services can be more easily made available to the user, such as betting, chat, and additional information about the race.

Figure 4.11 illustrates a possible configuration of the main elements organized into a web page containing: the game, the IPTV, the social networking elements, and other services.

Figure 4.11: Possible configuration of the elements into a web page

Note that the figure is used for indicative purpose only, and it does not necessarily reflect the final prototype.

The chosen application elements allow experimenting on different migration aspects. For example, the chat provides a significant case for experimenting migration on a widespread tool with multifaceted interface, the betting simulates a possible payment system, and the fast-paced game involves time-constrained migration issues.

In the following, each application element will be described in more detail, taking into account the simplifications that may be needed in order to cope with the complexity of the whole system.

4.2.4.1. IPTV

The implementation of a real IPTV service would be too demanding for the aim of the project. Moreover, licences for real races data are not easily obtainable and are out of the scope of the project as well. For all these reasons, a simulation will be considered. The simulation may be a recorded race of the game or a recording of a real race, converted in a suitable video format. Together with the video, telemetric data needed for the game may be exploited. The video could be transmitted in streaming to the web page, while the telemetric data could be used by the game in order to synchronize the game with the reproduction of the real race.

In a real IPTV, more services are expected to be available to the user, like channel selection, EPG and additional contents related to the race. As stated before, since no IPTV platform will be available, all these services have to be simulated as well: the channel selection and the additional content could be simulated by loading other videos locally or remotely available, while the EPG could simply be a navigable UI with simulated TV programs data.

4.2.4.2. Chat

The chat is a useful service which allows the communication among users that have the race as a common interest, and it is supposed to interact with some of the other services available. In fact, beyond the common features of a typical chat, such as the buddy list with user states (offline, online, etc.); additional information about the user presence should also be available. For example, the IPTV channel the user is watching at a given time, or the information whether the user is playing the race or not. Moreover, the chat service could be available on mobile phones, too.

4.2.4.3. Betting

The betting service allows the user to bet on the live race results. For example, she/he could bet on the pilot who performs the best lap, the winner of the race and so on. The main information available to the user are tipsters, bet parameters and bet results. Moreover a payment system should be available, supporting both credit cards and prepaid SIM cards. When using a prepaid SIM card, a SMS confirmation may be needed when the user is not using a mobile phone.

Since the implementation of the whole betting system goes beyond the scope of the project and the race is not real, a simulated version of the service will be realized. The GUI will be fully interactive and the data will be realistic, but without any connection to a real betting system, and the payments will be a simulation of real transactions, without connection with a real payment system.

4.2.4.4. Additional content

Some additional content related to the live race may be available. For instance, there could be information about track best times, gold book, and so on. Also in this case, the content will be realistic and navigable, although not directly related to the live race.

4.2.4.5. Game

The game will be a simplified prototype of a 3D racing game in which the players run their cars in a track. To be suitably integrated with the other application elements, it will be an embedded object in the web page. The game might be played against other pilots while watching a simulation of a live race on the IPTV.

The design of a racing game involves several elements, like car, tracks, game rules, and so on. The design of the cars has started from real F1 cars examples, by changing some features, in order to be more original and not to cope with copyright problems.

Graphic details like tyres, driving mirrors, and stabilizers, have been also taken into account. Starting from these sketches, some car models will be realized. Figure 4.12 shows an example of such sketches.

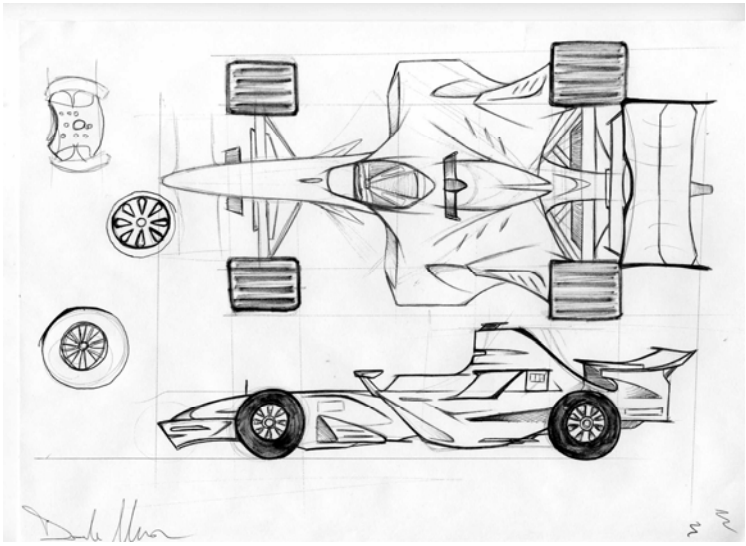

Figure 4.12: Example of car design sketches

The interior of the car has been studied too, with the realisation of preliminary sketches of the cockpit and main controls, as shown in Figure 4.13.

It is worth noting that the game does not pretend to be a realistic simulation, thus mechanical and aerodynamics issues have not been fully taken into account.

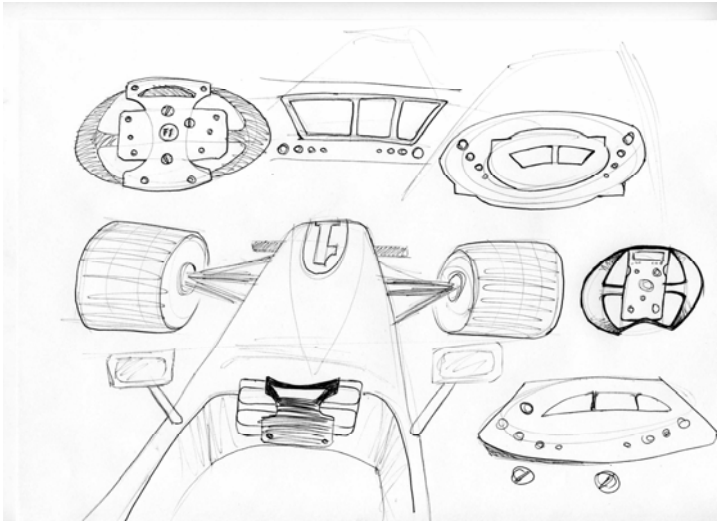

Figure 4.13: Preliminary sketches of car controls

4.2.4.6. Other functionalities

Since the application is supposed to run on a STB, a button for external internet navigation may be needed. As the application is a web-based one, a button that opens a new tab of the browser will be available to the user.

The different migration cases illustrated in 4.2.2 can be handled through a suitable user interface, which may be available as integrated in the web page. If the migration is started by the user, a list of available devices should be shown together with the list of application features supported by each device. Instead, in case the migration is triggered by the system, a user confirmation should be requested. Moreover the user could be able to activate or deactivate the automatic migration process.

These functionalities could be included in a specific area of the page, say a control area.

4.3. First year prototype

As mentioned before, the purpose of the application prototypes is to start making concrete experiments on migration technologies through an iterative approach. Starting from the requirements, some working lines have been identified that will gradually bring to working components, which will progressively be refined to allow a better understanding of possible migration issues. These working lines have led to the choice of developing two different applications. The first one can be considered as a starting point for early experiments on some migration functionalities, whereas the second one, being more complex, needs a more accurate design and analysis and it will be integrated afterwards with the platform.

The following paragraphs are dedicated to the description of what is expected to be realised for the first year prototype.

4.3.1. Arcade Game

In order to identify relevant aspects and technologies significant for migrating services and adapt them to the target platform, the Pacman game will be used as one of the first prototypes. It will basically demonstrate first solutions of migration of the user interface and adaptation of the application logic.

For the implementation the Model-View-Controller pattern (MVC) [2] has been followed, where the application is structured basically into three parts. One part is responsible for the interaction between the user and the game (*View*), another part manages the application logic (*Controller*), and the third part handles the application state (*Model*). These three parts are offering and accessing their functionality through services. Following this well-known software pattern, we are able to assign the techniques like application logic reconfiguration, state persistence, and migration of user interfaces to specific parts of the application.

The implementation itself will be based on Java and JavaScript, where the application logic part (Controller) will be implemented in Java, and the View will be realized using JavaScript. For the first prototype, the Controller and Model will run on the server side, while the View will reside on the client device.

The idea of the prototype is to migrate the View from a PC to a PDA. The View will then adapt its appearance to the resources available in the target device (screen size, CPU, etc). In addition, the application logic will be adapted in a way that takes into account the device limitations: for example the player gets higher score for catching ghosts because of the more difficult steerage of the Pacman. In a second step, a second player can join the game and can take control of one of the ghosts. That again results in reconfiguring the View as well as the Model and the Controller.

In order to adapt the game appropriately, context information like screen size and CPU rate are required. This will be done by using a context management framework introduced in deliverables D1.2 and D3.1. The various application services like View and Controller will communicate with each other using a middleware, which is based on CORBA (Common Object Request Broker Architecture) [3]. That allows us to abstract from network details like handling IP packages etc.

At the end, the prototype will show how different technologies like migration, application logic reconfiguration, and context management can be integrated and applied to an application in the game domain. Furthermore, the simple structure and implementation of the Pacman game itself allows focusing on developing and integrating those technologies, while we can omit functionalities of more complex games like physics engine and complex collision detection for example. Finally, we will gather valuable experiences in applying and integrating these various technologies.

Regarding UI migration aspects, a single desktop web page (written in XHTML and JavaScript) including different parts (see Figure 4.14), has been considered: the maze of the game with the different characters (the ghosts and the pacman), a part devoted to visualising the current state of the game (the number of pacman

lives still available, the current level of the game, the score), and interaction elements for controlling the game. With the latter wording we mean not only the controls for moving the pacman (the 4 arrow -imagemap, used for selecting the direction of the pacman) together with the two controls for the game (starting a new game, implemented with the central 'New Game' button in Figure 4.14, and pausing the game, which is implemented with the central round-shape button in the imagemap), but also the settings for specifying the general configuration of the game (e.g.: the animation speed of the pacman, the possibility of using random maze layouts for the various levels, etc.). It is worth pointing out that, in the desktop web page, for controlling the pacman, it is also possible to use the keyboard (e.g.: by using the arrow keys or some other predefined keys). In addition, in order to pause and start a new game also two keys are available ('n' and 'p', respectively).

When migrating to the mobile device, the original page will be split into two pages (see Figure 4.15): indeed the more limited capabilities of the target device do not allow to include all the elements in one page, also because the maze is a 14x20 table and then, even if it has been shrunk in order to fit the screen of the mobile device, it will occupy almost all the first cell phone page (see left-part of Figure 4.15).

Another aspect is the fact that, in the desktop version, the imagemap used for controlling the pacman is visualised in the same presentation in which also the maze of the game is rendered: this is done in order to allow the player to have an instant feedback of the selected direction, to control the position of the pacman, and to be able to react promptly when a ghost is approaching. This has to be preserved also in the mobile version, since a splitting in which the pacman controls are in a page different from the one where the maze is visualised will result in an unusable game (the user should navigate between different pages for controlling the game and check the updated situation in the maze).

However, as we said before, it is worth pointing out that in the current desktop version of the game there are two possibilities for controlling the game (through the keyboard and also by using the imagemap). Then, since a non-touchscreen mobile device is used as a migration target, the keyboard controls are considered by the adaptation engine more suitable than the imagemap. Therefore, the imagemap is removed by the adaptation engine, and it will not appear anymore in the mobile version.

Nevertheless, a possible disorientation problem might arise if a user was using the graphical imagemap for controlling the game on the desktop, and then, after migrating on a non-touchscreen mobile device, the imagemap does not appear anymore. In order to make the player aware of the change in the controls, a message in a pop up window will temporarily appear in the migrated interface on the mobile device, informing the player that the game has to be controlled from that point onward only through the keyboard. Such a message will display an image showing the numerical keys that have to be used on the mobile phone for controlling the game. In addition, when migrating to the mobile device, some controls are redesigned in order to be adapted for the mobile device: for instance, radio buttons with several choices will be replaced by pulldown menus, in order

to save space on the target device. Apart from visualising the maze, the first page will also contain the button allowing the user to start a new game and another button allowing the user to navigate to another page that will be available on the target device, which will include all the remaining elements (current pacman lives, current level and score, settings for animation speed and smoothness).

Regarding the state of the game, it should be preserved and then re-activated in the mobile device at the point where the migration was activated. Therefore, the current pacman and ghost positions should be saved, together with the current level and score and all the other settings that the player already selected in the desktop version.

Regarding more technological aspects, the original XHTML 1.0 code of the pacman game will be transformed in order to obtain an adapted version for the mobile device in XHTML Mobile Profile language. This process will be carried out by using a reverse engineering module written in Java which will reverse the original page and obtain an XML-based concrete description of the page. Then the semantic redesign module (implemented in Java as well) will support the adaptation rules allowing obtaining a new concrete description for the target device. Such a concrete description for the mobile device will then be updated with the values of the state, saved through AJAX-based scripts, which have been included by a proxy server in the considered page the first time the user asked an access to such a page. The final implementation for the mobile device, starting from the concrete description just mentioned, is implemented by a module written in Java as well.

In summary, the OPEN platform has to provide several functionalities to realize the described scenario. First of all it has to provide a communication infrastructure to enable communication between all involved components and services independent of the languages used for implementing them (e.g. Java or JavaScript). Secondly, the platform must be able to integrate different kinds of devices ranging from PCs to more resources-limited PDAs. Context information has to be gathered and analyzed in order to decide how to adapt the application appropriately. For instance, in the case of pacman application, an aspect that has to be taken into account when splitting the desktop page for obtaining the two adapted pages for the mobile device is the fact that the elements for controlling the pacman movement should be available in the same presentation where the maze is visualised, in order to allow the user to easily play the game. And finally, the platform has to provide the functionality of migrating the application or parts of it to any kind of device around, which also includes the capability of saving, just before triggering the migration, the state of both the application logic (e.g.: the JavaScript functions and variables) and the user interface, in order to preserve them on the target device.

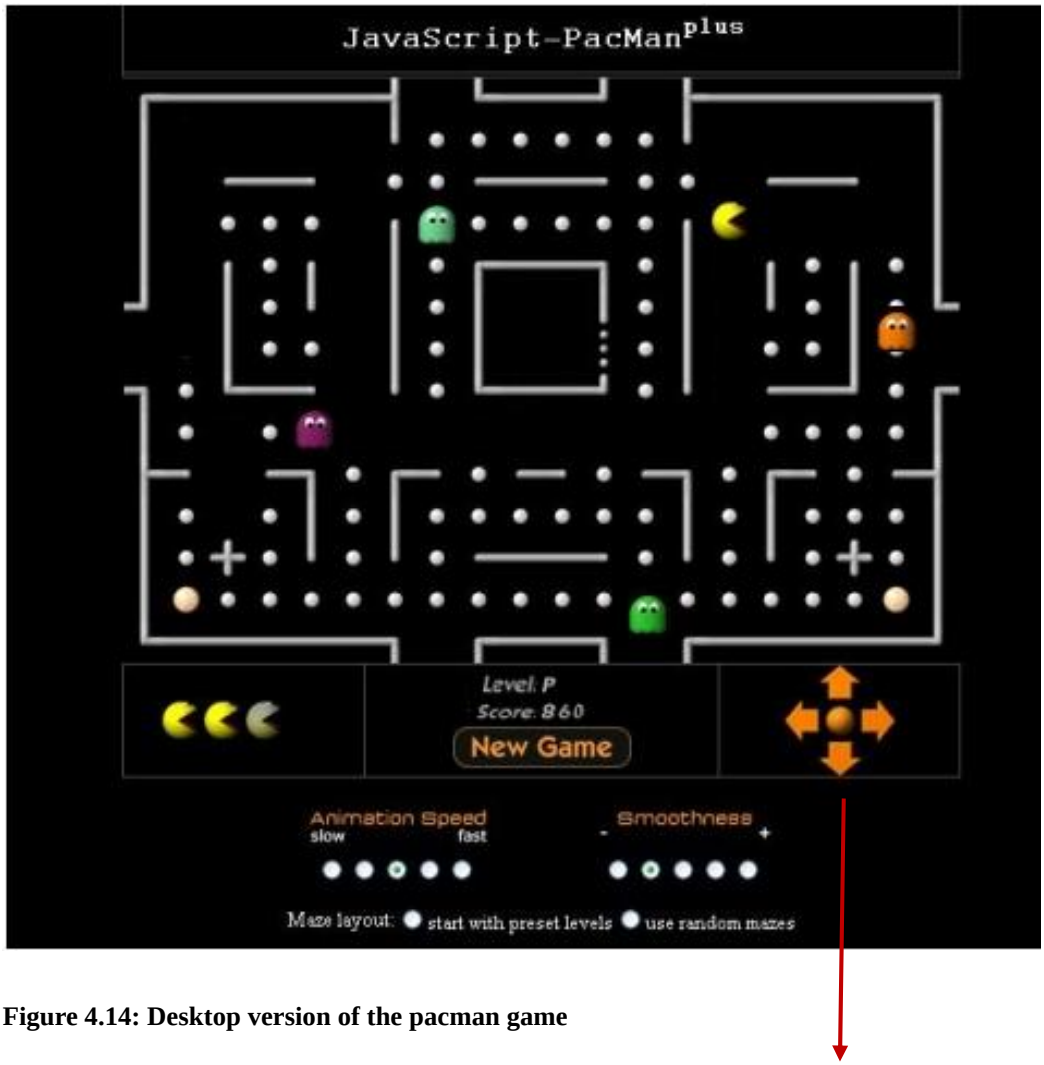

Figure 4.14: Desktop version of the pacman game

Imagemap for moving the pacman (4-direction arrows) and for pausing the game (central circle)

Figure 4.15: The two pages generated after migration for a mobile, non-touchscreen device

4.3.2. Social Game

The testbed application for experimenting on some of the migration issues arising from the modified IPTV gaming scenario will be realised through a static web page containing some of the elements described in the previous paragraphs. The elements do not interact with each other, but a few of them (or a part of) could be separately migrable.

The web page will be divided into a number of areas, each of them being dedicated to a specific functionality. In particular, there will be an area dedicated to display the IPTV, one for displaying additional content, one for chatting, another one for betting, one large area for playing the game, plus a set of controls for handling with the migration and internet browsing.

The **IPTV** will be simulated using one or more videos (eventually locally stored) that could be watched by the user and controlled through classic buttons like play, stop, channel, etc. The area in the page will be implemented through a Shockwave-Flash application.

The **Chat** will be an adapted version of a JavaScript open source chat.

The **Betting** functionalities will only be simulated but an interaction with realistic and navigable information will be considered.

The area dedicated to **Additional content** will contain navigable information about the simulated race, with elements like images, lists and texts.

The control area will basically contain two buttons: *migration*, devoted to open a menu for simulating migration options, and *internet*, devoted to open a new tab for free internet browsing.

The **Game** will be implemented in C++ and hosted in an embedded browser object through a native plug-in. To avoid platform-related issues, this version of the plug-in will only support Firefox browser running under the Linux operating system. The game will use OpenGL as 3D libraries. The game will be a simplified prototype of a multiplayer 3D racing game with a predefined maximum number of players. This first version will follow a Time Attack modality, which means performing the best lap in the shortest time. Players can see each other during the race and collide among them.

Before entering the game, a player sets his preferences in a settings file external to the game. When he enters the game, a car is automatically assigned to him, with a colour different from that of the other players. The player can enter the game in any time during the race, and he starts from the pit lane. A numbered position in the pit lane is assigned to him according to the queue of entrance. After the formation lap, his time is measured once he crosses the finish line for the first time.

During the race, the best lap time, together with the name of the player who made it, will be always visible. The player can leave the race whenever he wants, by pressing the exit key. Two different points of view will be available, one with a camera following the car, the other one from inside the cockpit of the car.

Another aspect that will be taken into consideration in the first year prototype will be the design of the prototype UIs, for the devices involved in the migration. This will help on the one hand to select the most suitable UI according to the device, and on the other hand on improving UI migration adaptation strategies and techniques. Some sketches that will help to experiment layouts in different devices will be produced and made available for usability evaluation.

5. Conclusions

In this deliverable, the guidelines for defining the prototype applications that will be developed during the course of the project have been described and discussed. This evolving reference document, which will be finalized in D5.3 - Final application requirements and design, should be considered as a tool for communicating the vision of the applications to the consortium, in order to share among all the partners what the application prototypes are supposed to be.

The document has been organized into three sections: the first one describing the main goals and motivations, the second one and the third one dedicated to the two selected application domains, namely the business domain and the game domain, respectively.

In the current version, only the game domain has been described. The business domain will be covered in another release of this deliverable.

Regarding the game domain, games that support migration among different devices are not available to date. Starting from the scenarios analysed and proposed in D1.1, together with the corresponding migration cases, two different types of games experiences, which represent complementary set of requirements, have been considered. The first ones are Arcade games, which combine a well known game experience with a not very complex application; the other ones are Social games which present novel game experiences and a very complex application. The latter may find in migratory services a fundamental element for improving the overall user experience. Besides analysing possible game experiences, some technical investigation about the feasibility of specific application components, have been also made.

After a preliminary analysis on the impact of the arcade and social game experiences, the complex application arising from social gaming has been considered as a case study for defining an ideal scenario for investigating migration aspects, eliciting requirements, and making an early design.

Finally, the prototypes that are expected to be realised in D5.1 have been described.

6. References

[1]VRR (http://www.volere.co.uk/index.htm).

[2] Fowler, Martin. “Patterns of Enterprise Application Architecture”, Addison-Wesley, 2003.

[3] Object Management Group. “Common Object Request Broker Architecture”, January 2008, version 3.1. Online available at: http://www.omg.org/docs/formal/08-01-04.pdf