Multi-Dimensional Context-Aware Adaptation of Service Front-Ends

Project no. FP7 – ICT – 258030

Deliverable D5.2.3 Application Prototypes (R2)

Due date of deliverable: 30/09/2013

Actual submission to EC date: 30/09/2013

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
Dissemination level		
PU	**Public**	**Yes**

Document Information	
Lead Contractor	SAP
Editor	SAP
Revision	
Reviewer 1	CNR-ISTI
Approved by	Telefónica I+D
Project Officer	Michel Lacroix

Contributors	
Partner	**Contributors**
SAP	**Atul Vaibhav, Safdar Ali, Joerg Rett**
W4	**Jean-Loup Comeliau, Nicolas Bodin**
TID	**Javier Caminero, Mari Carmen Rodríguez**

Changes			
Version	**Date**	**Author**	**Comments**
0.1	03/07/2013	SAP	First draft send to partners
0.2	16/07/201	TID, W4	Partner content input
0.6	12/09/2013	SAP	Consolidated version
0.7	19/09/2013	CNR-ISTI	Reviewed version send to SAP
0.8	23/09/2013	SAP, TID, W4	Final version sent to cons. lead
1.0	30/09/2013	TID	Final version sent to EC

Executive Summary

This deliverable presents the third and final release of the Serenoa application prototypes with a focus on their implementation. The document describes the three scenarios, proposed by the three industrial partners in which the prototypes are deployed. The E-Health scenario of TID, the E-Commerce scenario of W4 and the Warehouse Management scenario of SAP are three independent scenarios, for which the prototypes are being developed.

The prototypes use the Serenoa concept of Reference Models (T3.1) to define the adaptation use-cases. The adaptation rules are described according to the Serenoa language for adaptation rules (T3.3) using the authoring tools (T4.5). These rules are uploaded to an Adaptation Engine and triggered through the events from the Context Manager (T4.4). The abstract model according to the Serenoa language (T3.2) has been used at design time while the Runtime Generation Engine (T4.1) produces the final prototype interface. Finally the prototypes are evaluated within task T5.3-Evaluation.

The focus of this description will be on the implementation of use cases and adaptation rules. Making use of the Serenoa framework technical solutions, the prototypes are continuously gaining fidelity and functionality. The final state of the prototypes and the implemented integration into the Serenoa framework are reported in this deliverable.

Table of Contents

1 Introduction

1.1 Objectives

The main objectives of work package WP5 are threefold:

- Task T5.1 'Generic Integrations' addresses the integration of technologies for context-aware SFEs developed in the main scientific and technological work packages (WP2 to WP4).
- Task T5.2 'Prototype Development' has investigated three prototype applications based on technologies developed in the previous work packages.
- Task T5.3 'Evaluation' has evaluated the functionality, usability, effectiveness and quality of the developed technologies from a practical perspective.

In this third and final phase of task T5.2 'Prototype Development' the focus is on development of the application prototypes, as well as the integration of the Serenoa framework developed throughout different work packages into the current prototypes. The descriptions of the prototypes were updated including the adaptation use cases and the implementation details.

1.2 Audience

This document has a public dissemination level, so theoretically it is open to public consultation by the general public. However, a key audience is represented by Project reviewers and officer, as well as any researcher/scientist who could be interested in the topics addressed by Serenoa.

1.3 Related documents

- Deliverable D5.2.2 is the second version of "Application Prototypes" deliverable.
- Deliverable D1.1.2 Requirements Analysis (R1) is the latest deliverable which describes the scenario's requirements and evaluation criteria.
- Deliverable 5.1.2 Serenoa Framework describes the technical modules developed within the Serenoa project
- D3.1.1 "Reference Model Specification" provides an exhaustive list of possible adaptation use cases, which are partly taken over from the scenarios as described in section 2.

1.4 Organization of this document

After giving a formal introduction in section 1, section 2 presents the scenarios which are the basis of the development of the prototypes. The main contribution of this deliverable is presented in section 3, where the technical realization of the current prototypes is explained, adaptation use cases are listed and the integration of the Serenoa framework is demonstrated. Section 4 summarizes the contributions of the deliverable and section 5 gives directions for future work.

2 Scenarios

In this chapter we will summarize the scenarios of the Serenoa project. Compared to previous deliverables, the description of the three scenarios has been updated. As argued in the previous release of the deliverable [9], such changes in the scenarios are not uncommon, since scenarios are living documents which can be modified at any point of time in order to adapt to contextual changes or to reach a more detailed or even more abstract level.

Each industrial partner in Serenoa has designed an own application scenario upon which the prototypes will be developed. These scenarios were already presented in detail in the previous release of the deliverable [9] and will be updated in this document. The three scenarios are:

- The E-Health scenario from TID
- The E-Commerce scenario from W4
- The Warehouse Management scenario from SAP

The final prototypes emerging from these scenarios and our expectations concerning the user experience when interacting with them will be presented later in section 3.

2.1 E-Health Scenario

The scenario for the TID's Serenoa prototype is aligned with TID's E-Health strategic area, where an important motivation is to provide medical patients with tools and applications so that are were able to remotely perform certain medical interactions. Thanks to the Serenoa prototype, a multi-device access to E-Health applications can be provided and users will be able to interact with them using their preferred devices. More details on the related pilot programs can be found in the previous version of this deliverable [9].

A significant driver for the progress of the E-Health field is to obtain a significant reduction in the chronic patients' management costs by using online self-assessment software, such as the one employed in the Chronic Patients scenario, in order to diminish the huge economic impact on the health system that the tens of thousands of these chronic patients pose. Additionally, the patients will also be satisfied with these remote management tools because they reduce their need to be personally in touch with the health system, and their quality of life is in this way improved.

Since the beginning of the project, different market changes have occurred, so the initially dominant position of Microsoft Windows as preferred Operating System is changing and now in the new smartphones and tablets, Android OS has become the most widespread option. Therefore, Telefónica has adapted its initial strategy for allowing a successful future exploitation of Serenoa results, focusing its efforts on creating a version of its prototype running over Android, which can help to show the multi-device capabilities of the Serenoa framework.

As compared to the E-Health Scenario in the previous version of this deliverable [9], the scenario has received some modifications:

Maria, an elderly woman with a chronic disease, uses daily the SARA interface (a 'Chronic Patients' application) in order to check her medication and doctor's appointments. The avatar is in charge to offer an alternative way of navigation through the system, although Maria is allowed to hide this component at any time, if desired. She has some visual problems, so the system reads aloud the indications and the reports sent from her doctor. At the beginning of the interaction, the system proposes Maria to measure her blood pressure since it has been requested by her doctor. She starts getting her blood pressure but she does not remember very well the procedure. Therefore, the avatar offers her a demo video showing her how it works. This explanatory material is enriched with links to external content (e.g. Wikipedia) in order to clarify the technical terms or the meaning of certain parts of the process.

2.2 E-Commerce Scenario

The goal of the W4's Serenoa prototype is the implementation of an E-Commerce scenario that illustrates various features that are adaptive to the user's context. Software solutions dealing with Multimodality and Accessibility are explored. In this scenario we illustrate how the application content and operations can be made understandable, with predictable behaviours, by different profiles of users. We also illustrate multimodality by showing how navigation and screen content can be controlled by head position and gesture when the application is used on a device equipped with a Web cam.

The scenario was originally defined by a business process with typical exchanges between a front office application (for customers) and a back office application (for the seller). The real world use-case for this scenario is inspired by a company that earns revenues from keeping a fraction of the transaction amount and serves as an intermediary between the product providers and the customers. More details can be found in the previous version of this deliverable [9].

This scenario illustrates differences in the application's behavior depending on the context of use: if it is accessed by a standard user (with no identified disabilities) or by a user whose profile requires software adaptive features (based on device, language or disability).

We also illustrate dynamic adaptation of screen content based on SERENOA adaptation rules describing how the application behaves depending on head position or gesture:

- When the user's head is detected as near the screen, the screen automatically shows the details of a catalogue's item. If the user moves his head backward, the list view of the catalogue is displayed.
- When the user's head moves to the right, the next page (or the details of the next item) is displayed; when it moves to the left, the previous page (or the details of the previous item) is displayed automatically.

This adaptation may be relevant to update the way information is displayed on the screen: when the user nears his head, one may argue that he intends to see more details, and the nature of the information displayed can be updated.

2.3 Warehouse-Management Scenario

In order to ensure a just-in-time and just-in-sequence deployment of relevant components to the respective assembly stations, many companies define a proceeding consignment step (see [9] for details). A possible scenario for such a warehouse picking process has been described in the previous version of this deliverable [9].

As compared to the Warehouse-Management Scenario from the previous version of this deliverable [9], the scenario has received minor modifications:

The usage of a scanner unit has been removed as this functionality can be solved with the camera of a mobile device or the next generation of Head-Mounted Displays. As an additional modality besides vocal and visual we have introduced vibro-tactile feedback to support the adaptation process. Due to this additional feature, we have replaced the attachment of the mobile device from the belt to the arm to allow the vibro-tactile feedback. Finally, the language localization functionality has been taken out as it is regarded as an already available standard feature.

3 Design of the Prototypes

This section is divided into three parts corresponding to the three application prototypes.

The respective section in the previous version of this deliverable [9] started with describing the adaptation use cases partially taken from the deliverable D3.1.1 “Reference Model Specification” [12]. This document presents an update, e.g. additional adaptation use cases that have emerged.

The existing integration of the Serenoa project (i.e. use of specific modules) will then be presented to demonstrate how the technical solutions developed in other work packages can be used for real applications. An overview on the architecture is given in Figure 1 and Figure 2 .

Finally, the specific technical realization of the current prototypes is described.

The Serenoa architecture represented by the adaptation data flow shown in Figure 1 consists of three main components. Firstly, at design-time, ‘*Authoring tools*’, which are available in both client-based and web-based version, are used to support the design of model-based user interfaces. Secondly, the ‘Theoretical framework’ consists of reference models, a design space and an ontology. Finally, the run-time phase will transform the description and associated rules of user interfaces into a final user interface implementation. A more detailed description can be found in deliverable D.1.2.2 “Architectural Specifications (R2)” [13].

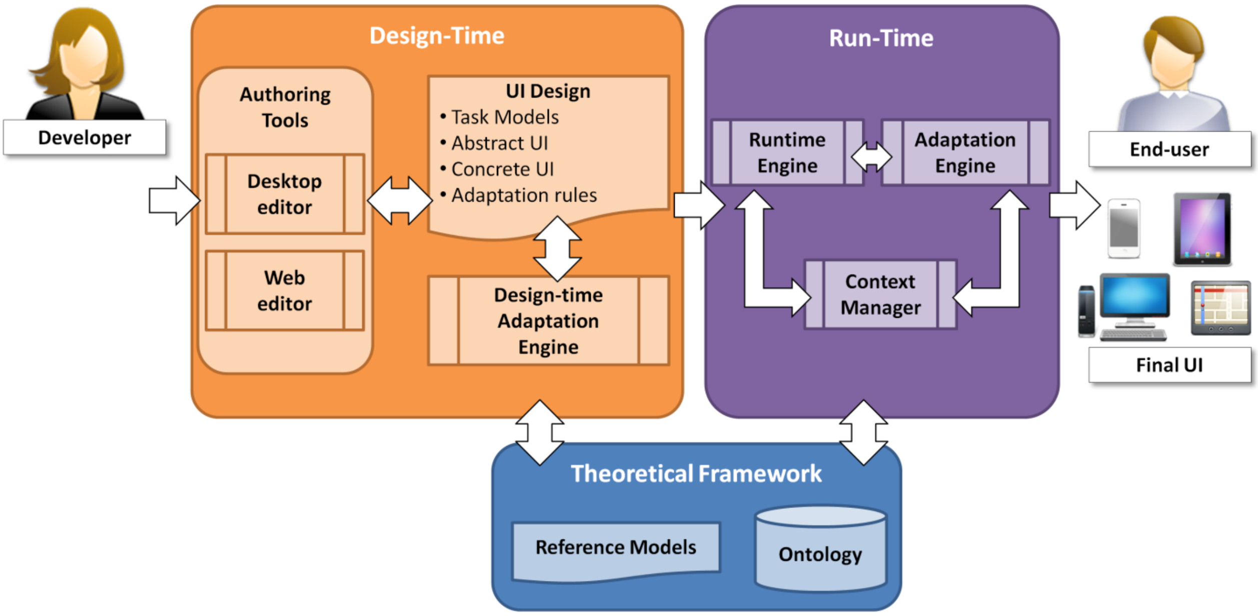


Figure 1: Serenoa architecture represented by the adaptation data flow [13].

Figure 2 shows the Serenoa architecture from a different viewpoint but relates to the same components. A more detailed description can be found in deliverable D.5.1.1 Serenoa Framework (R1) [14].

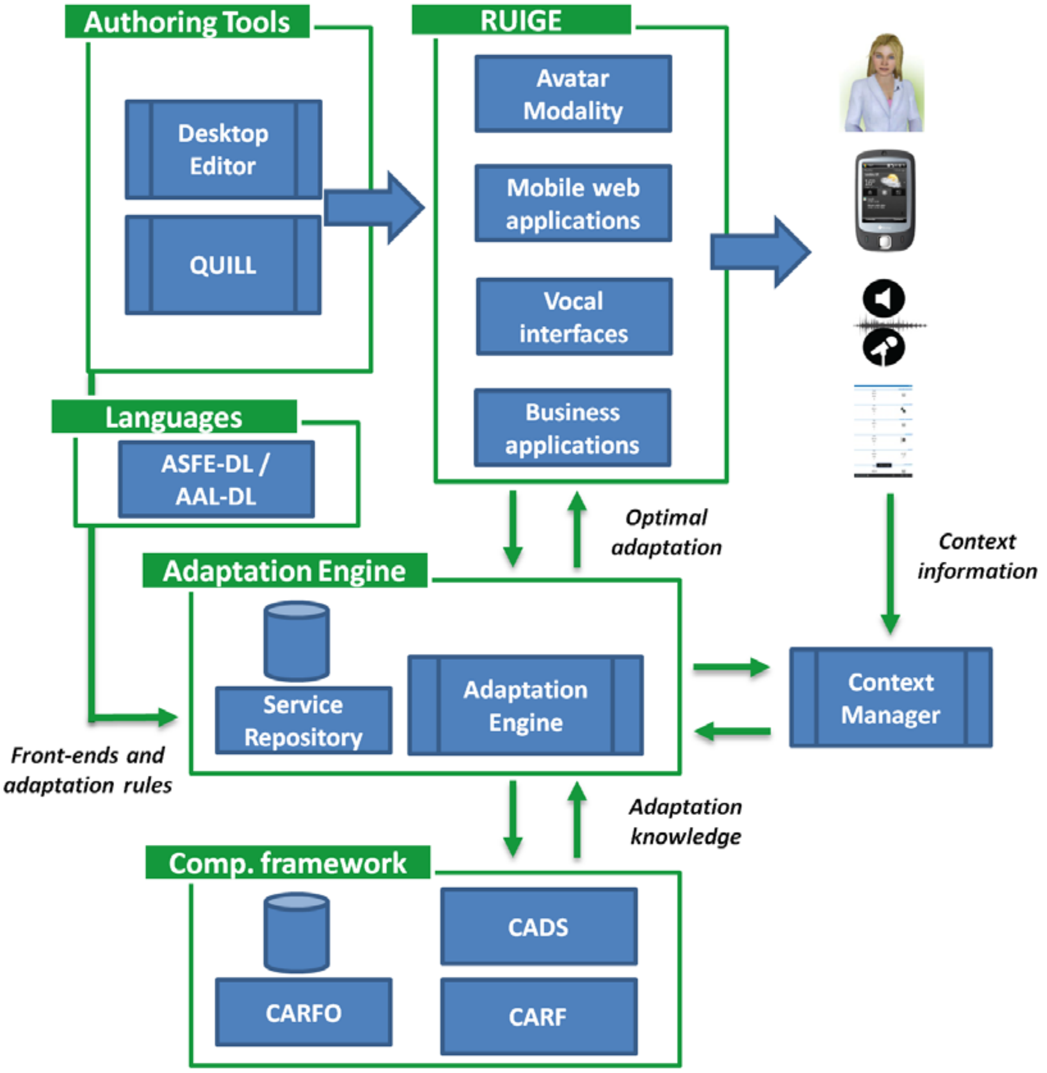


Figure 2: Serenoa architecture [14]

3.1 E-Health scenario

The E-Health scenario is aimed to demonstrate the use of the Serenoa framework in a chronic disease patients-oriented application (SARA), using adaptive ECAs (Embodied Conversational Agents) in order to enrich the interactions and creating also a version running on a state-of-the-art Android-based tablet.

3.1.1 Adaptation use cases

The different scenarios illustrating the Intelligent-Picking prototype are described through the following adaptation use-cases. Those use-cases that have already been explained in previous release of the deliverable are shown in a bullet list. A more detailed description can be found in previous release of the deliverable [9]:

- Adaptation to Translate Language
- Adaptation to Re-size
- Adaptation to Snapshot
- Adaptation to Suggest
- Adaptation to Change Volume
- Adaptation to Translate Modality
- Adaptation to Alternative Navigation

- Adaptation to Stretch
- Adaptation to Translate to Audio
- Adaptation to Direct Guidance (e.g. Suggest, Recommend)
- Adaptation to Change of Attitude
- Adaptation to Frame-based
- Adaptation to Annotation of Links

3.1.2 Integration of the Serenoa framework

We propose the following example of adaptation (see Figure 3) in order to show the integration and role of the different Serenoa's components into the E-Health scenario. The adaptation process is performed through the Serenoa framework as follows: Jane starts using the desktop application, at home. When she is outside and tries to access the E-Health assistant using her smartphone, the Runtime module (RUIGE) is in charge to degrade the avatar presentation and present the sequence-of-images version. At a certain moment, the Context Manager notifies about a high level of noise, thus it becomes useless having the avatar speaking. So, again Runtime module is responsible to automatically stop the avatar voice feature.

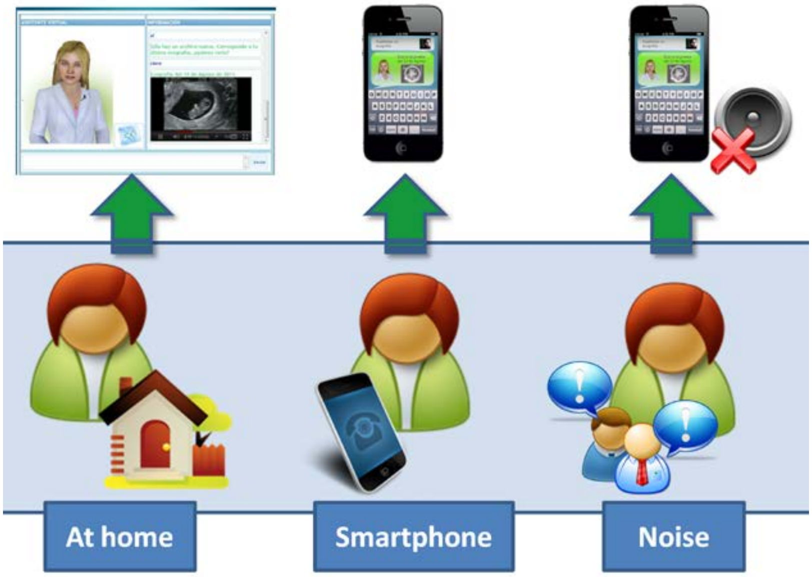


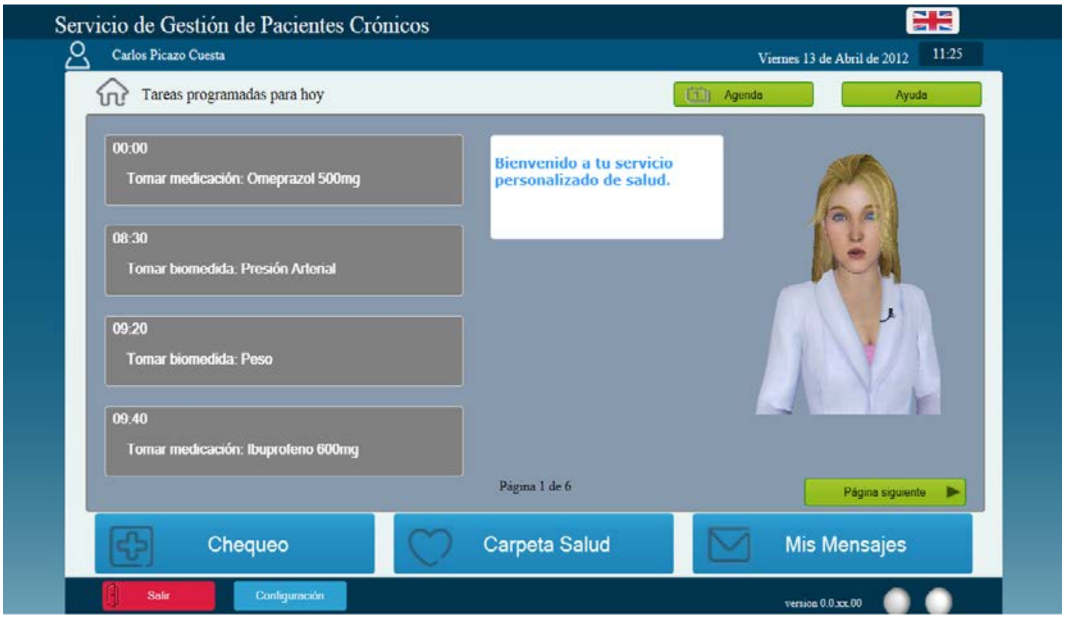
Figure 3: Adaptation example for E-Health scenario

3.1.3 Technical Realization

The E-Health scenario is aligned with some E-Health projects from TID. The technologies employed in the implementation of the interface had to be changed or adapted to include the Haptek[1] avatar engine and to make it flexible enough for a smooth integration into the Serenoa framework.

Thus, our first implementation effort has been the development of the interface mock-ups corresponding to the considered scenario. In Figure 4, the main interface of the SARA project is shown. In the left part (grey boxes), the daily tasks are listed (e.g. the pills or tests that the doctor has programmed). In the right side, the avatar has been embedded and although it offers text-to-speech capabilities, a textual version is also displayed in a text area (i.e. the white one in the middle of the interface). The buttons located at the bottom and at the top of the interface give access to the remaining application functionalities (i.e. the agenda, the tests, the inbox, etc.)

[1] http://www.haptek.com/



Servicio de Gestión de Pacientes Crónicos

Carlos Picazo Cuesta

Viernes 13 de Abril de 2012 11:25

Tareas programadas para hoy

00:00 Tomar medicación: Omeprazol 500mg

08:30 Tomar biomedida: Presión Arterial

09:20 Tomar biomedida: Peso

09:40 Tomar medicación: Ibuprofeno 600mg

Bienvenido a tu servicio personalizado de salud.

Página 1 de 6

version 0.0.xx.00

Figure 4: Snapshot of the SARA project interface

The development of the prototype has been performed using HTML5, CSS and JavaScript libraries as jQuery[2]. For the integration of the avatar, Haptek provides an ActiveX component which is embedded as an object in the code of the interface. The control of the behaviour of the avatar (i.e. expressions, speech, etc.) is performed through different JavaScript commands.

The SARA's user interface requires a minimum screen resolution of 800x600 pixels and the UI elements such as buttons, text boxes and menus are placed in accordance with this prerequisite. Moreover, the Avatar engine is an ActiveX component so it can properly run only in Internet Explorer over Windows platforms.

At the moment, for desktop and laptop computers, Microsoft Windows is the most widely adopted Operating System and Internet Explorer is one of the most used browser in those environments. For that reason, the SARA project has been developed and tested on those platforms.

However, if we move to mobile and tablet environments, Android and iOS are the most popular platforms, although there are more mobile-oriented platforms such as Blackberry, Windows Phone, Bada and webOS. The desktop user interface of SARA project presents some usability issues in mobile and tablet devices; therefore Serenoa project allows the possibility of performing the required adaptations to run SARA in those environments.

In 2012, web traffic from traditional desktop computers has decreased around 10%, in favour of mobile web traffic (tablets and mobile phones). Web traffic has dropped from 88% to 79% and mobile web traffic has increased from 10% to 20%. The tablet and smartphone use nearly doubled in that year. It is expected that in following years the use of mobile technologies will continue increasing.

[2] http://jquery.com/

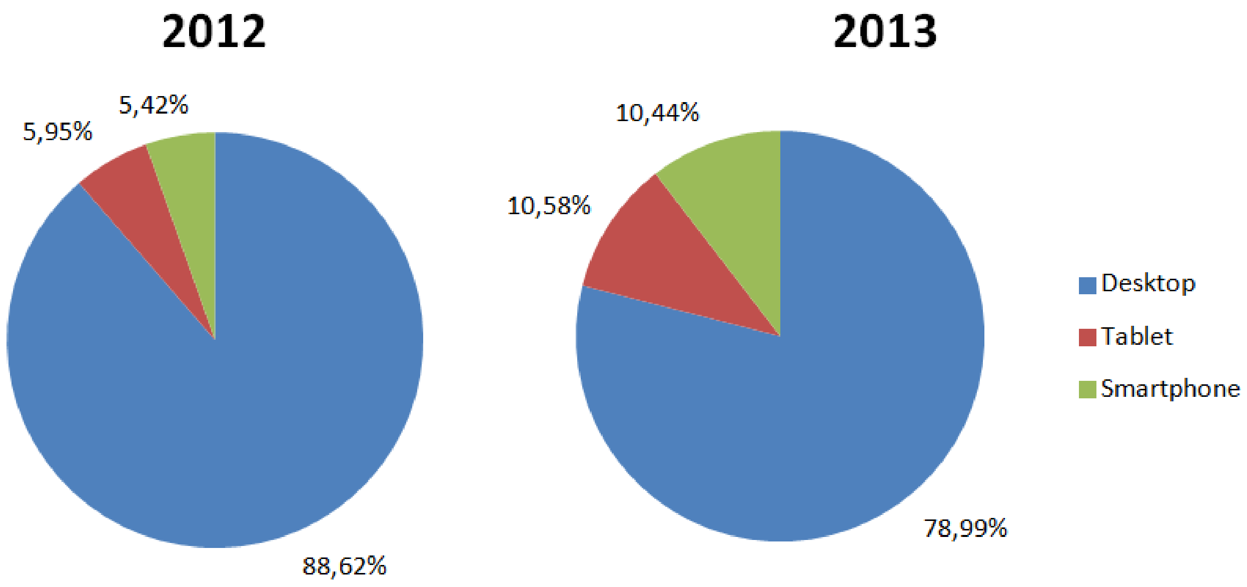


Figure 5 - Device share of web traffic

This trend is very significant. Nowadays, applications should offer a mobile interface as well as a traditional interface if they want to reach the broadest user population. Therefore, a mobile adaptation of the Serenoa's E-Health prototype is needed, so the user can use the application whenever and wherever s/he wants.

There are several mobile platforms in the mobile market scene. Providing implementations for all of them is very difficult because each platform has its own features. So, we focused only in some of them. Europe's market share shows iOS and Android platforms as the most used ones nowadays.

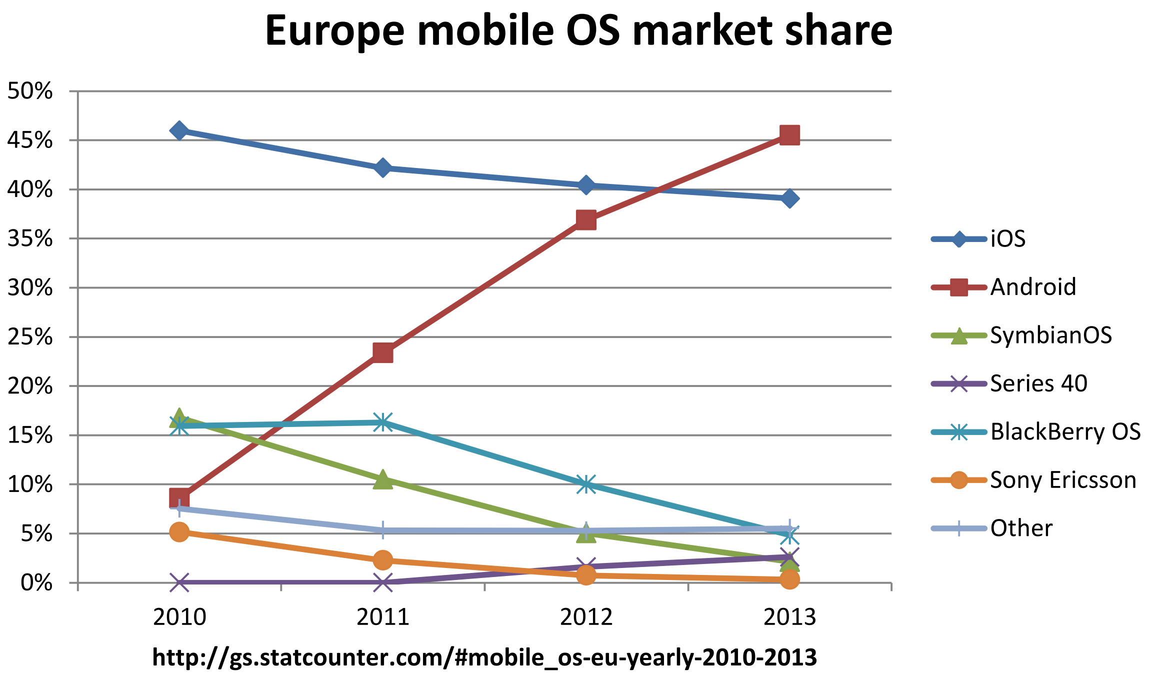
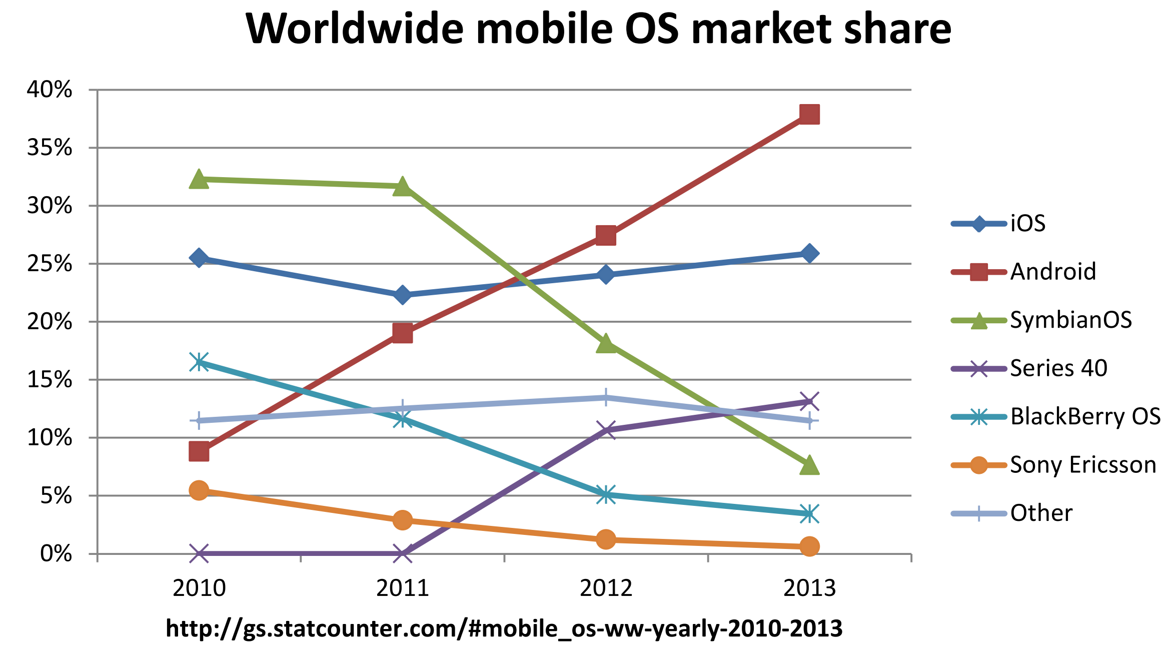


Figure 6 - Europe mobile OS market share

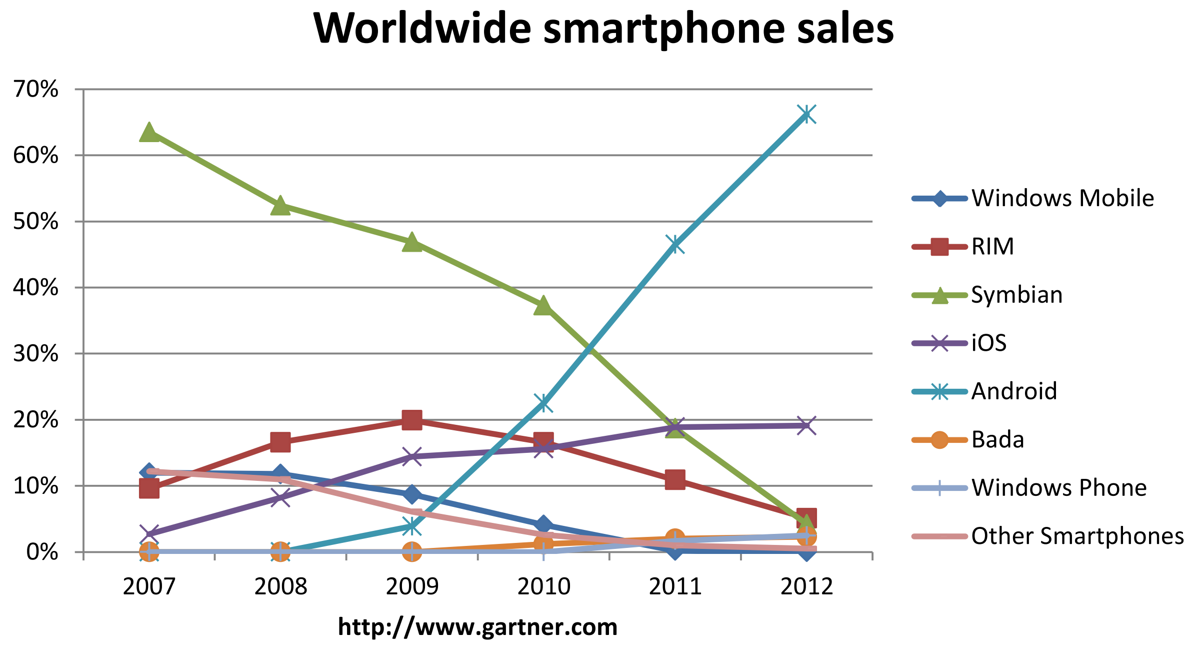
While the iOS market share has maintained constant, Android has increased a lot since 2010, beating iOS in 2012 in Europe. This trend is not only a European trend. Taking a look to the worldwide mobile market share, we can observe the same behaviour.



http://gs.statcounter.com/#mobile_os-ww-yearly-2010-2013

Figure 7 - Worldwide mobile OS market share

We can explain this behaviour if we take a look to the smartphone sales. It is clear that, since 2009, Android devices have been sold on a massive scale, in contrast to Symbian and Blackberry devices.



http://www.gartner.com

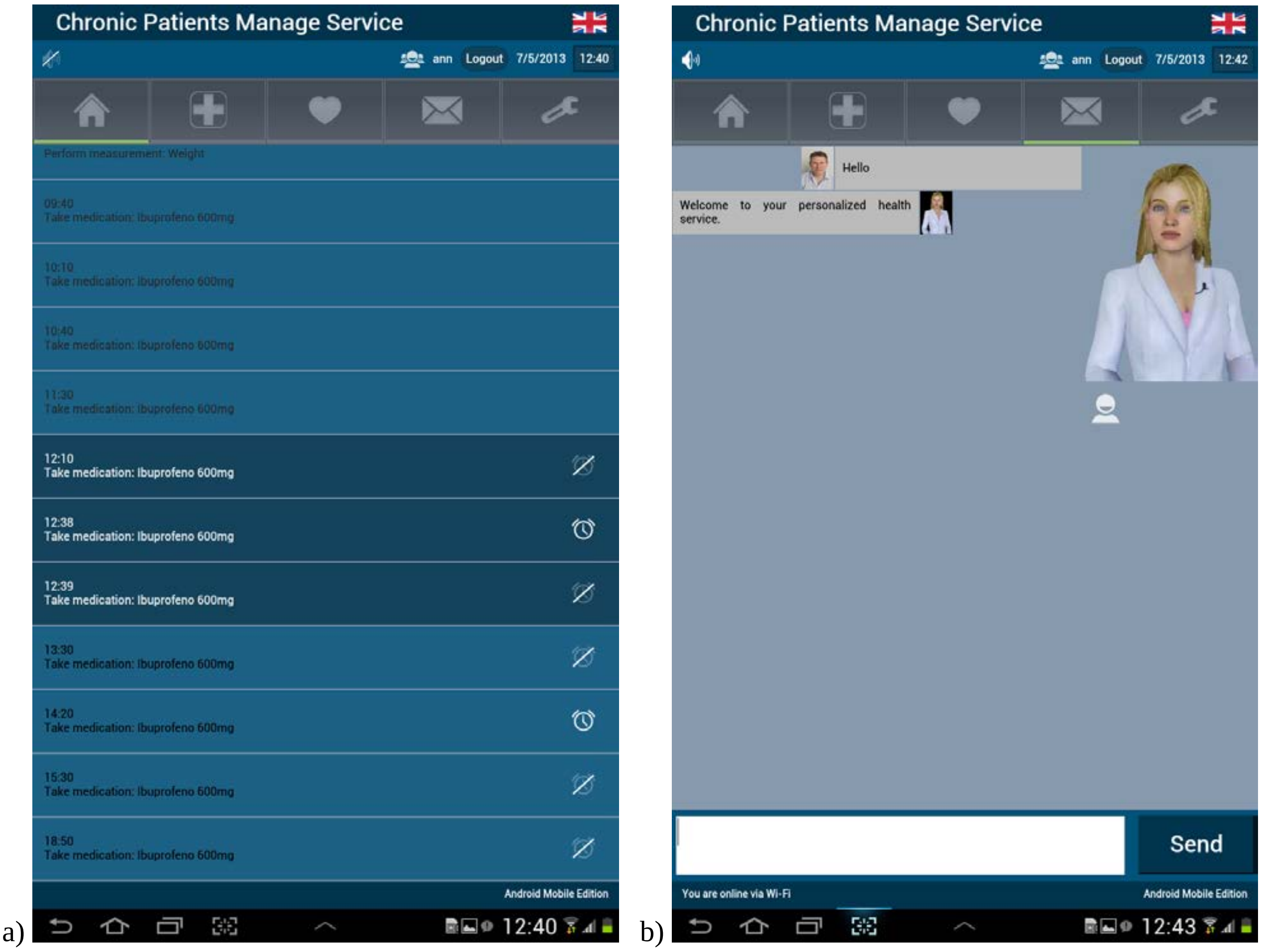
Figure 8 - Worldwide smartphone sales

It is thus reasonable for mobile applications to focus mainly on Android and iOS platforms.

In order not to develop different applications for each mobile platform and not to waste resources, it is desirable to use some kind of library that isolates the development from the individual implementations in each platform. Phonegap[3] (also known as Apache Cordova) provides implementation for several mobile platforms and allows the development of an HTML5 application capable to run in those different frameworks. Using this framework gives us the benefits of developing a web application for Androd, iOS, Windows Phone, Blackberry and Bada using HTML5, CSS3 and JavaScript. This allows a faster creation process which is also easily adaptable to different mobile Operating Systems.

Trying to adapt the E-Health prototype to these devices, a mobile-device application has been developed using the Phonegap/Crodova framework. We have selected an Android device to show the demo, since Android devices are currently the most sold platform worldwide as shown in Figure 8.

Figure 9 a) shows the main screen of the SARA project adapted to mobile and tablet devices.



a)

Chronic Patients Manage Service

ann Logout 7/5/2013 12:40

Perform measurement: Weight

09:40
Take medication: Ibuprofeno 600mg

10:10
Take medication: Ibuprofeno 600mg

10:40
Take medication: Ibuprofeno 600mg

11:30
Take medication: Ibuprofeno 600mg

12:10
Take medication: Ibuprofeno 600mg

12:38
Take medication: Ibuprofeno 600mg

12:39
Take medication: Ibuprofeno 600mg

13:30
Take medication: Ibuprofeno 600mg

14:20
Take medication: Ibuprofeno 600mg

15:30
Take medication: Ibuprofeno 600mg

18:50
Take medication: Ibuprofeno 600mg

Android Mobile Edition

12:40

b)

Chronic Patients Manage Service

ann Logout 7/5/2013 12:42

Hello

Welcome to your personalized health service.

Send

You are online via Wi-Fi

Android Mobile Edition

12:43

Figure 9: a) Main screen of Android application b)

Features of the desktop application have been ported to the mobile one. The language and avatar preferences of the application are retrieved from the Context Manager. If the user preferences state that the avatar must be shown, we place a video of the avatar instead of the previously mentioned ActiveX component (which is not suitable with Android applications).

The dialogs of the avatar have been implemented as chunked videos. We make use of the capabilities of the <video> HTML5 element (see Figure 9 b)) and the *currentTime* property, which allows us to fast-forward or to back-ward to certain instance of the video, providing quick access to particular snippets of action. Moreover, the avatar can be hidden or shown pressing a button, updating the new user preference to the Context Manager.

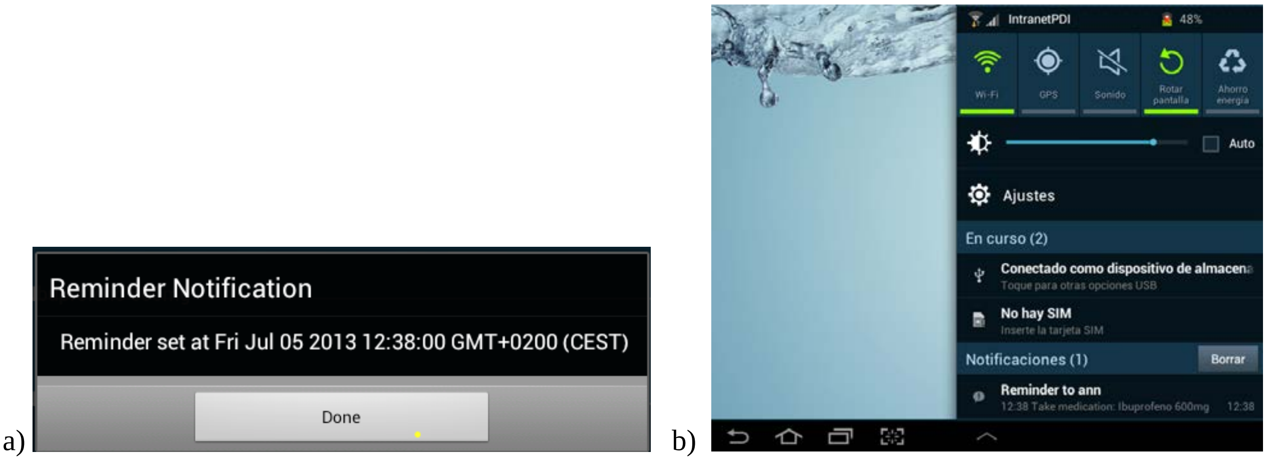
The application also incorporates text-to-speech capabilities making the avatar more user-friendly and

[3] http://phonegap.com/

improving accessibility. This functionality is provided by the integrated Android text-to-speech engine thanks to Phonegap/Cordova framework, which makes it possible to use Android native components from the web application.

The third adaptation implemented is the one driven by the noise level. If the noise level surrounding the user is low enough, then the text will be read by the avatar. When the noise level is higher, it is preferable not to read the text, because the user may not be able to hear it. So, the avatar is muted. A context delegate application continuously senses the noise level detected by the device microphone and sends the value to the Context Manager. When some relevant event occurs (e.g. when the noise level exceeds a predefined threshold), the Context Manager fires an asynchronous notification to the Serenoa Framework and the RUIGE adapts the text-to-speech functionality. As an example, in Figure 9 the loudspeaker icon (top-left) indicates that output speech is muted because high noise level has been detected. In contrast, in Figure 9 b) low noise level has been detected and output speech is enabled.

In addition, the application has a notification service to remind the user which actions s should be performed as recommended by the doctor, such as taking pills and visiting the doctor. As the application makes use of Android notification service, all notifications will be available although application is not open, as shown in Figure 10 a) and b).



Reminder Notification

Reminder set at Fri Jul 05 2013 12:38:00 GMT+0200 (CEST)

Conectado como dispositivo de almacena
Toque para otras opciones USB

No hay SIM
Inserte la tarjeta SIM

Reminder to ann
12:38 Take medication: Ibuprofeno 600mg 12:38

Figure 10: a) Date details when set up a reminder notification b) Notifications even when applications is closed

3.2 E-Commerce scenario

The e-commerce prototype is an e-commerce scenario for a company which sells bicycle related products on a public website. The main access for customers is a public web site (the application front end).

When customers place orders using their **front-end**, a workflow activity is initiated in the company for employees to validate the order, check payment, ship the products and track the order until the customer has received and checked the content. Therefore, the second module is a **back-end** application, which is mainly an intranet business application dedicated to employees and managers of the company.

3.2.1 Adaptation use cases

In this section, we present a list of foreseen adaptations in the context of the e-commerce scenario. Such adaptation use cases are a sample of the different adaptations that will be offered by the prototype.

The different scenarios illustrating the E-Commerce prototype are described through the following adaptation use-cases. Those use-cases that have already been explained in previous release of the deliverable are shown in a bullet list. A more detailed description can be found in previous release of the deliverable [9]:

- Adaptation to the user’s language (French/English)
- Adaptation to the user’s disability (colorblind or not)
- Adaptation to the user’s device:
    - Desktop device with HTML5 browser

- Desktop device with a standalone (Java/Swing) application
- Mobile device with HTML5 browser
- iOS native application, using a specific software component, the iOS Player
- Android native application, using a specific software component , the Android Player

In addition to these adaptions two new adaptation use-cases have been introduced which allow a dynamic adaptation to head position and gesture. For these adaptation use-cases the detailed description will be used:

Use Case Name	**Screen content adaptation based on user's head position**
Use Case Description	A scenario for a shopper (front-end user): A web visitor connects to the bike shop using his desktop and browses through the items. When the distance between his head and the screen (computed using a web cam) varies, the window is adapted: - To show the details of the item when the distance narrows - To show a more global view of the catalogue (list item view) when the distance widens This case shows dynamic adaptation to the user's position
Primary Actor	English speaking shopper using a desktop.
Precondition	The desktop must be equipped with a webcam and the browser must be webRTC compatible. The user visualizes the catalog of items for sale at the bike shop. The initial distance from head to screen is computed and serves as a reference for further adaptation to know whether the user is getting closer or further to the screen.
Trigger	The web client is informed that the user has moved his eyes closer to the screen or further to the screen
Basic Flow	Based on the current content of the screen, the adaptation rule switches to some different content showing either the details of an item (in case distance decreases), either a list view of the catalog (in case distance increases)
Alternate Flows	Same features available on mobile devices equipped with frontal cameras and using a compliant browser (supporting webRTC)

Use Case Name	**Screen content adaptation based on user's head movement**
Use Case Description	A scenario for a shopper (front-end user): A web visitor connects to the bike shop using his tablet and browses through the items. When his head is detected as moving to the right (detected using a web cam), the window is adapted: - To show the details of the next item when the current view shows the details of an item - To show the next page of the catalogue when the current view shows the list view of items This case shows dynamic adaptation to the user's movement
Primary Actor	English speaking shopper using a tablet
Precondition	The tablet must be equipped with a webcam and the browser must be webRTC compatible The user visualizes the catalogue of items for sale at the bike shop.

	The initial head position is detected and serves as a reference for further adaptation to know if the user is moves it to the right or to the left.
Trigger	The web client is informed that the user has moved his head to the right or to the left
Basic Flow	Based on the current content of the screen: - If the current content shows the details of an item, then the adaptation rule switches to the next item details (head moving to the right) or to the previous item details (head moving to the left) - If the current content shows a page of items, then the adaptation rule switches to the next page (head moving to the right) or to the previous page (head moving to the left)
Alternate Flows	Same features available on desktops equipped with a webcam and using a compliant browser (supporting webRTC)

The last version of the prototype differs from the previous one in that:

- It implements the latest adaptation rule (content adaptation to head gesture and position)
- It includes an optimized communication with SERENOA"s adaptation engine

3.2.2 Integration of the Serenoa framework

The adaptation process performed by the Serenoa framework is as follows: the shopper starts using the desktop application at home. When he leaves and reconnects from his smartphone in the bus, the Runtime module (RUIGE) is in charge to adapting presentation to his new context. If a backend user (customer representative or manager) connects, then preferences (language, color-blindness) are also taken into account by the RUIGE. The runtime module is responsible for recognizing the type of device that is being used.

An automatic adaptation of the catalogue view is also automatically decided by the RUIGE when relevant (if a webcam is available and webRTC supported by the browser). In this case, the Generator module changes the screen content to adapt to the user's position and movements.

As of today (July 2013), the integration of a SERENOA Context Delegate for headtracking remains to be implemented. The current implementation works at a client level only (JavaScript), but to comply with the SERENOA architecture notifications to update the context Manager need to be sent by a Context Delegate. This mechanism will have to be tuned based on performance results because, in this case, it would be more natural for adaptation to take place in the client only.

The two kinds of adaptations dealt with in Serenoa are considered in the e-commerce prototype:

- **Runtime adaptation**: these rules apply when a user requests for content: the runtime generation process of the page takes the rule into account and sends back the adapted content. These rules are stored within the service warehouse and there is no need to restart the application when the rule changes. For instance: change language, content adaptation for color-blind persons, restore user's profile, filter information...

- **Design time adaptation**: these rules impact the way the application behaves, independently of how the end-user behaves after he starts using it. Design time rules are therefore used by the adaptation engine to propose a modified version of the application. A new version of the application can then be created and deployed.

The following diagram shows the overall E-commerce prototype architecture:

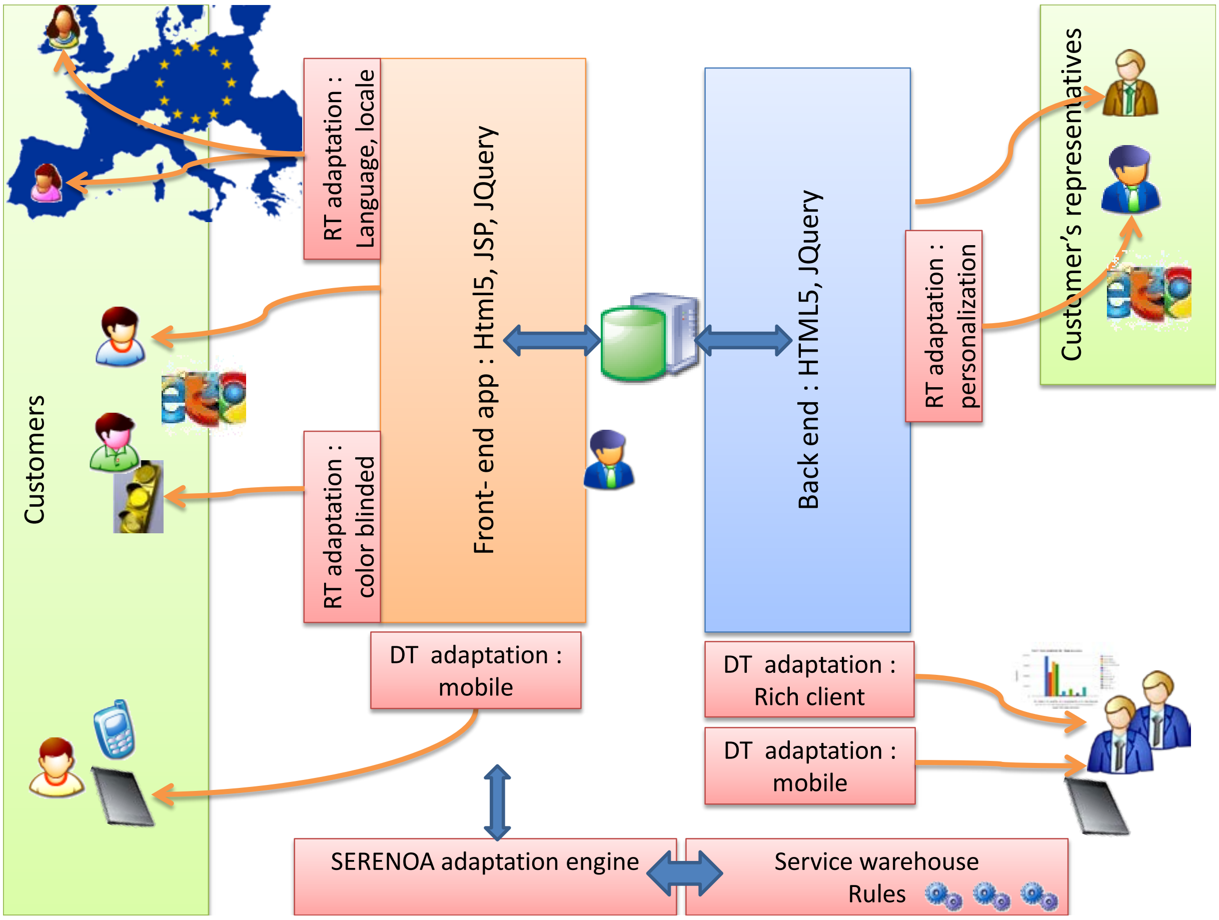

Customers
RT adaptation : Language, locale
Front- end app : Html5, JSP, JQuery
RT adaptation : color blinded
DT adaptation : mobile
Back end : HTML5, JQuery
RT adaptation : personalization
Customer's representatives
DT adaptation : Rich client
DT adaptation : mobile
SERENOA adaptation engine
Service warehouse
Rules

Figure 11: E-Commerce Prototype Architecture

InFigure 11, the different adaptations implemented in the E-Commerce scenario are illustrated. There are both design time adaptations (DT) and runtime (RT) adaptations.

The scenario mainly interacts with 2 Serenoa modules: the adaptation engine and the service warehouse, where rules are stored. The scenario will support various types of connections:

- On the web (HTML5/jQuery for desktop view) for both employees and customers
- As a rich client application (java/Swing) for employees
- On mobile terminals: both iOS and Android, Smartphones and tablets for both employees and customers

3.2.3 Technical Realization

Front-end features

On the front-end (the public web application to see products, prices and place orders), the “customers” can browse across products, and may possibly create purchase orders and then follow up with the shipping process.

In our scenario, some people may access the public website from various countries and expect to find navigation or product description either in English or in their native language.

Naturally, with web deployed applications, some actors will want to connect using a browser from their mobile (Smartphone or tablet) device. But this may lead to some unfriendly UI, where interactions with the end-user may be difficult and even deter customer loyalty. Therefore, offering a more appropriate UI based on native apps (making use of the device’s proprietary graphical components) is a relevant adaptation of the application for the mobile world, and may help improve brand image.

Marketing teams may request some ‘flashy’ graphical effects to have an attractive public website: however some color-blind users could be disoriented. An adaptation to such disabilities could also open the internet website to a wider public.

Back-end features

On the back end, the employees validate, follow, and answer questions about incoming orders from customers. Unlike the customers who connect periodically to the public website but for a limited time during the day, employees work all day long with the application. They expect to find more functionalities to adapt the content of the back-end application to their profile: they need to be able to get rid of useless information, change graphical style like colors and fonts. They also expect to be able to save and retrieve their own profile upon login.

Finally, managers need to have a global view of the business activity and usually work with a large amount of orders: they require some business intelligence capabilities such as charts, reports, data filtering. One possibility is to propose a rich client interface for such managers where they will be offered more advanced graphical capabilities and a faster access to data.

When out of the office, managers also might be interested in displaying key indicators in dashboard views on their mobile terminals, for performance analysis for example. Therefore, some subset of their Business Intelligence (BI) features must be offered.

Figure 12depicts how the E-Commerce scenario is technically realized:

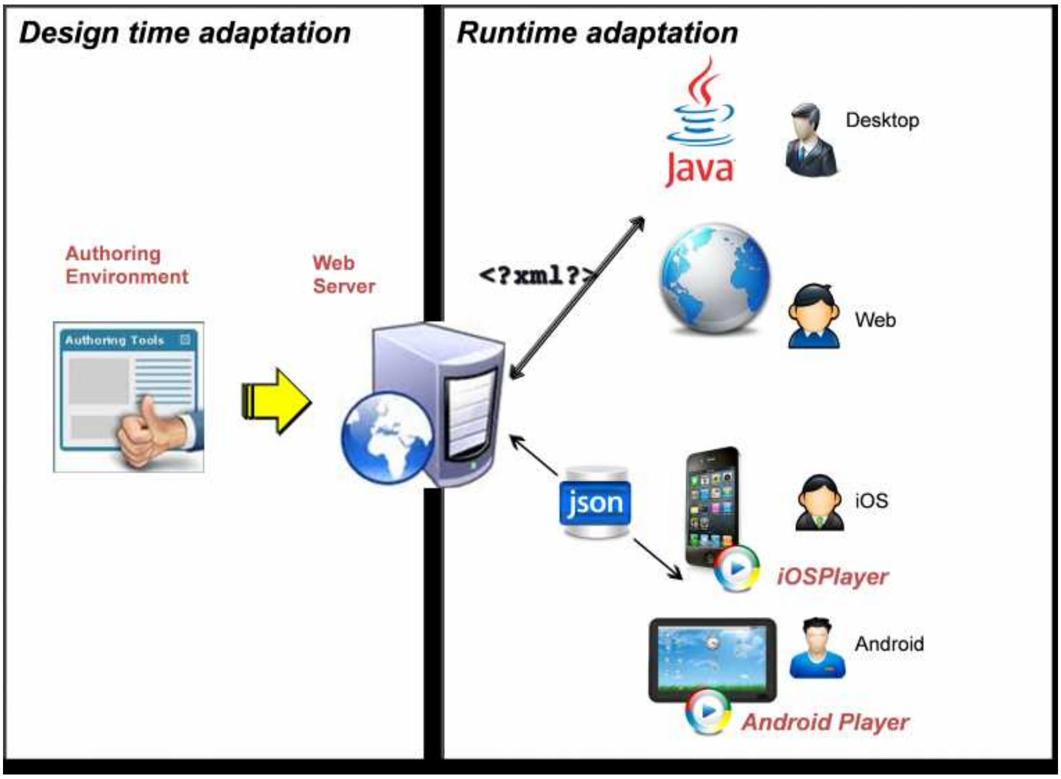


Figure 12: Technical Realization of E-Commerce Scenario

At design time, rules can be set to adapt the display based on static factors. This first set of rules is taken into account to generate a war file that is deployed on a web server. At run time, the web server waits for incoming requests from different types of users.

The four types of users illustrate how content considered are:

- The **desktop user**: typically a manager who wants to use BI oriented feature for supervising purposes.
- The **web user**, who uses his desktop browser to visit the bicycle shop: typically an online shopper or

an employee (acting as a customer representative, for instance).

- Mobile users, including **iOS user**s and **Android users**.

The technical components involved for this scenario include:

- **Native application players for both iOS and Android platforms**. These pieces of software need to be installed on the mobile terminals to render the application as interaction occurs through the JSON protocol.
- **The Java** display manager, using the Swing environment, exchanging XML documents with the server (in this case, a web server is not needed).
- **The web2.0** display manager, using the HTML5/jQuery environment, exchanging XML documents with the web server

As for face detection (needed for adaptation rules requiring face movement detection), we used *headtrackr*[4], a JavaScript library for real-time face tracking and head tracking, tracking the position of a user's head in relation to the computer screen via a web camera and relying on the webRTC[5]/getUserMedia standard.

3.3 Intelligent-Picking Prototype

The objective of the Intelligent-Picking prototype (IPP) is to demonstrate of the possibilities obtained through adaptive user interfaces. So, instead of focus on business logics of system, what we describe is the implementation of these adaptive rules for the UI introduced above. Comparing to the traditional interaction, in this prototype the UI is fully audio-based: the users are hand free when they interact with it. However, the associated problem is the imperfect speech recognition and the burden of memorizing audio commands, which make audio interface not always able to suit for interaction between users and system. Therefore adding the adaptability to UI may make up for the deficiency of audio interface. The self-adaptive features enable user to finish the picking task more efficiently and effectively.

3.3.1 Adaptation use cases

In this section, we present a list of adaptation use cases which are all reasonable in the context of the warehouse management scenario. Not all of these adaptation use cases are implemented in the final version of the prototype.

The different scenarios illustrating the Intelligent-Picking prototype are described through the following adaptation use-cases. Those use-cases that have already been explained in previous release of the deliverable are shown in a bullet list. A more detailed description can be found in previous release of the deliverable [9]:

- Adaptation to a fragile item (minimize distraction)
- Adaptation to remind the next action (destination shelf reached)
- Adaptation to noisy environment (switch off vocal modality)
- Adaptation to a traffic jam (optimize path)
- Adaptation to an experienced worker (suppress information)
- Adaptation to change speed (Audio)
- Adaptation to translate language (Audio & Text)
- Adaptation to change brightness (all Content)
- Adaptation to change font family (Text)

3.3.2 Integration in the Serenoa framework

As in the previous release of the deliverable [9] we start by showing the integration of the Intelligent-Picking prototype (IPP) in the Serenoa framework through a content map (see Figure 13). The map shows the relationship between the IPP and other supporting modules of the Serenoa framework.

From the application scenario described in task T5.2-Prototype Development a set of adaptation use-cases

[4] More at https://github.com/auduno/headtrackr
[5] More at http://www.webrtc.org

can be defined using the patterns and format described in task T3.1-Reference Models. The resulting adaptation use-cases have been documented in the previous section 3.3.1. From theses use-cases we are able to describe the relevant adaptation rules in terms of events, condition, and actions according to the Serenoa language for adaptation rules as specified in task T3.3-AAL-DL using the tool developed in task T4.5-Authoring tool. These rules are uploaded to an Adaptation Engine and triggered through the events from the Context Manager developed in task T4.4-Context of Use. Those events can depend on various aspects (user preferences, environmental changes, application-related events …).

Switching to the left side of Figure 13, the adaptive features of the UI can be realized through the use of model-based descriptions of interactive applications. At design time the various initial versions of the applications are developed in terms of three concrete descriptions (vocal, graphical, and multimodal) in the abstract model according to the Serenoa language as specified in task T3.2-ASFE-DL. The generator based on the MARIA language, as defined in task T4.1-RUIGE, produces the final IPP interface. The IPP communicates with the Adaptation Engine to receive adaptation actions triggered by the Context Manager. Finally the IPP is evaluated within task T5.3-Evaluation

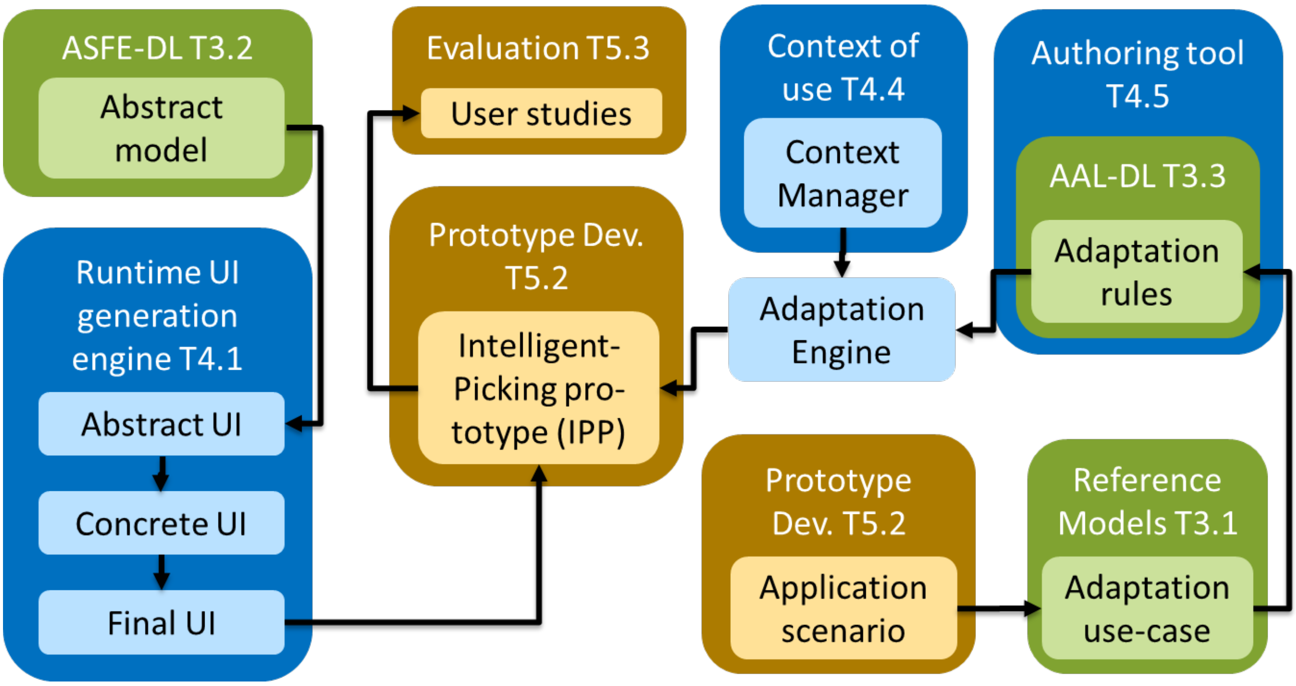


Figure 13: Content map of Serenoa modules around the Intelligent-Picking prototype. The colours green, blue and brown indicate the Serenoa workpackages 3 (Models, Languages and Methodology), 4 (Algorithms, Runtime and Tools) and 5 (Integration & Experimentation) respectively.

3.3.3 Technical Realization

In the previous release of the deliverable [9] the architecture was described as an UI component based on web-technologies which communicated to the Application serves containing the Adaptation Business Logic, Adaptation Rule Engine and the Context Manager. It was described that the multimodal interaction happens visually through the UI screen on the HMD and vocally through the mobile device (wearable computer). The adaptation was described as being requested by the Adaptation Engine (server) after being triggered by the Context Manager (i.e. a change in the context). It was foreseen that some changes might be triggered by tracking the picker’s position.

It can be verified, by looking at Figure 14, that these main concepts have remained, though they are depicted now in more detail and with some extension.

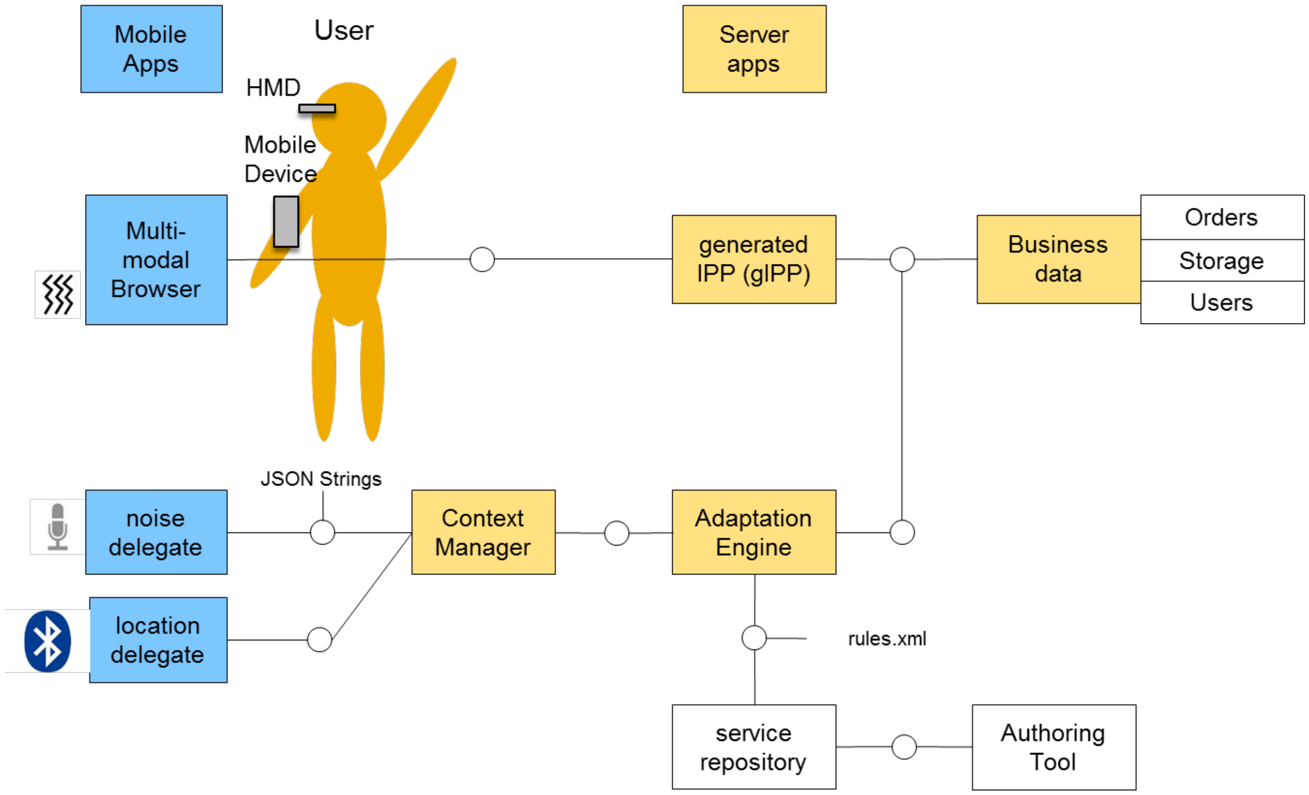


Figure 14: Adaptive prototype architecture

The IPP has now been generated using the Maria RUIGE (generated IPP - gIPP). The Business data is provided by RESTful services containing the data of the actual orders, the infrastructure of the storage (i.e. a map of the shelves) and the users. The Adaption Engine holds the rules in form of an XML scheme which can be created by the Authoring Tool. The Context Manager connects to two delegates running on the mobile device concerning noise and location. The multimodal browser which is running also on the mobile app allows voice out- and input and a vibro-tactile feedback.

Figure 15 shows the adaptation to minimize distraction when picking a fragile item as a sequence of screenshots and the respective vocal interaction. After reaching the destination shelf 2.02 (screen 1) a multimodal (vocal and visual) notification will be triggered (screen 2 – 4). After that the visual modality will be switched off. The confirmation will use only the vocal modality (screen 5). The following task will switch back to multimodality (screen 6).

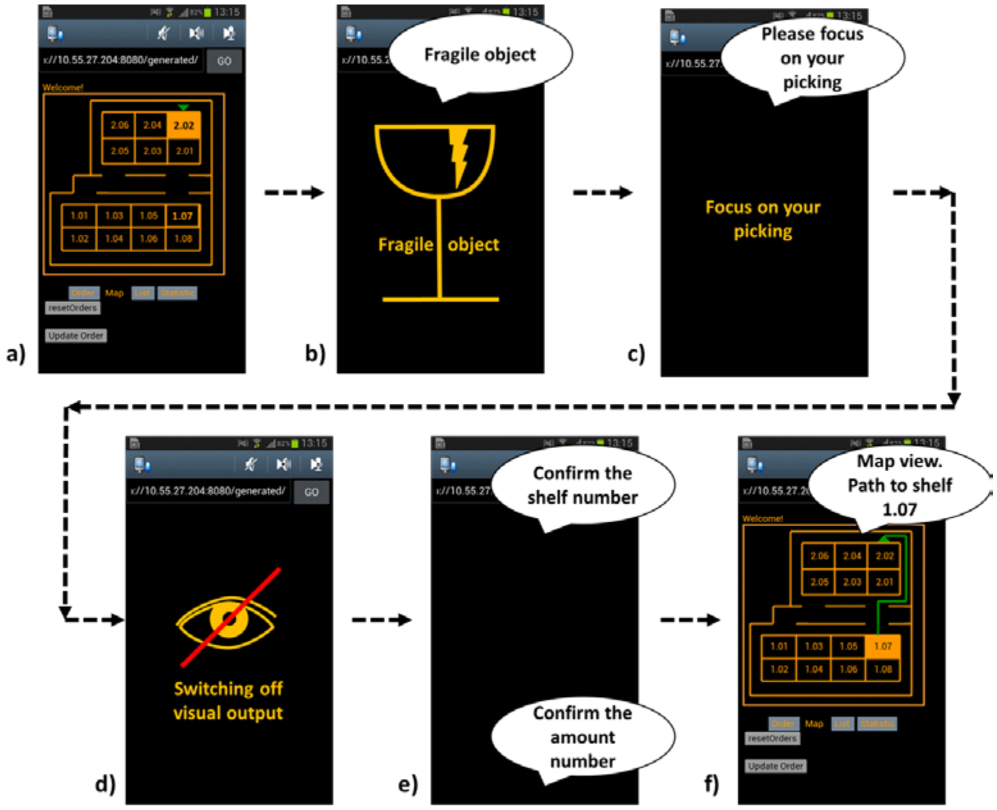


Figure 15: Adaptation to minimize distraction when picking a fragile item. Upon reaching the destination shelf (a) the notification appears and the visual modality is switched off (b – d). The confirmation will use only the vocal modality (e). The following task will switch back to multimodality (f).

4 Conclusions

This document reports the final status of the development of the Serenoa application prototypes with a focus on implementation.

The document follows three lines of works which reflect the three application prototypes developed by the tree industrial partners TID, W4 and SAP. The main topics are:

- the scenarios
- the status of the final prototypes
- the integration of the Serenoa framework in the prototypes
- adaptation use cases

The E-Health scenario of TID is aimed to demonstrate the use of Serenoa framework in a chronic disease patients-oriented application (SARA), using adaptive ECAs (Embodied Conversational Agents) in order to enrich the interactions and creating also an alternate multi-device version running on a state-of-the-art Android-based tablet. Trying to adapt the E-Health prototype to mobile devices, a mobile-device application has been developed using the Phonegap/Crodova framework. TID has selected an Android device to show the demo. Features of the desktop application have been ported to the mobile one.

The e-commerce prototype is an e-commerce scenario for a company which sells bicycle related products on a public website. The main access for customers is a public web site (the application front end). The technical components involved for this scenario include native application players, a Java display manager and a web2.0 display manager.

The objective of the Intelligent-Picking prototype (IPP) is to demonstrate more about its features of adaptive user interface. Comparing to the traditional interaction of user interface, in this prototype the UI is fully audio based that is the users are hand free when they interact with system. The IPP has now been generated using the Maria RUIGE which also indicated in the naming. The Business data is provided by REST services containing the data of the actual orders, the infrastructure of the storage (i.e. a map of the shelves) and the users. The Adaption Engine holds the rules in form of an XML scheme which can be created by the Authoring Tool.

The final versions will be evaluated regarding the requirements fulfilment, the extent of the integration of the Serenoa framework and the reasonability of the adaptation use cases. The prototypes will be used to drive the continuing efforts to promote the use of the adaptive UI technologies within the partners' organizations and their customers.

Summarizing, this document outlines the final state of the prototype development, shows how the Serenoa framework is integrated in their technical realization and presents scenario-specific adaptation use cases.

References

1. Berry, D. C.; Butler, L. T Rosis, F. (2005), 'Evaluating a realistic agent in an advice-giving task', *International Journal of Human-Computer*. & de *Studies* 63(3), 304-327.
2. López-Mencía, B.; Hernández-Trapote, A.; Díaz-Pardo, D.; Santos Cámara, R. & Rodríguez, M. (2008), *ECA gesture strategies for robust SLDS*, AISB 2008 Convention - Communication, Interaction and Social Intelligence on the Symposium on Multimodal Output Generation, Aberdeen, Scotland.
3. Rubin, J. (1994), Handbook of usability testing. Systems Technology, Wiley, 262
4. Schwaber, K., Sutherland, J. (2011), The scrum guide, Retrieved from http://www.scrum.org/
5. Weissenberger, U., Fellenz-Thompson, C. (2009) User-Centered Design. SAP User Experience, SAP AG. Retrieved from http://www.sapdesignguild.org/resources/ucd_paper.asp
6. Patton, J., Halley, L. (2009) Making sense of UCD and Agile. Presented at Computer-Human Interaction Forum of Oregon (CHIFOO) May 6, 2009. Retrieved from: http://www.slideshare.net/LaneHalley/making-sense-of-ucd-and-agile
7. Dickinson, J., Kumana, D. (2010) Agile and UCD: Building the Right Thing, the Right Way. devx Retrieved from http://www.devx.com/enterprise/Article/42156/1954
8. Losada, B., Urretavizcaya, M., and Castro, I. F. (2011). An integrated approach to develop interactive software. In Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part IV (INTERACT'11), Pedro Campos, Nuno Nunes, Nicholas Graham, Joaquim Jorge, and Philippe Palanque (Eds.), Vol. Part IV. Springer-Verlag, Berlin, Heidelberg, 470-474.
9. Serenoa Consortium (2012), Deliverable D5.2.2 Application Prototypes (R2). Available at: http://www.serenoa-fp7.eu/wp-content/uploads/2012/09/SERENOA_D5.2.2.pdf
10. Serenoa Consortium (2011), Deliverable 2.4.1 Criteria for the Evaluation of CAA of SFEs, Project co-funded by the European Commission within the Seventh Framework Programme. Available at: http://www.serenoa-fp7.eu/wp-content/uploads/2012/07/SERENOA_D2.4.1.pdf
11. Serenoa Consortium (2011), Deliverable 3.4.1 Agile Methodology Description (R1), Project co-funded by the European Commission within the Seventh Framework Programme. Available at: http://www.serenoa-fp7.eu/wp-content/uploads/2012/07/SERENOA_D3.4.1.pdf
12. Serenoa Consortium (2011), Deliverable 3.1.1 Reference Models Specification (R1), Project co-funded by the European Commission within the Seventh Framework Programme. Available at: http://www.serenoa-fp7.eu/wp-content/uploads/2012/07/SERENOA_D3.1.1.pdf
13. Serenoa Consortium (2012), Deliverable D.1.2.2 Architectural Specifications (R2), Project co-funded by the European Commission within the Seventh Framework Programme.
14. Serenoa Consortium (2012), Deliverable D.5.1.1 Serenoa Framework (R1), Project co-funded by the European Commission within the Seventh Framework Programme.

Acknowledgements

- TELEFÓNICA INVESTIGACIÓN Y DESARROLLO, http://www.tid.es
- UNIVERSITE CATHOLIQUE DE LOUVAIN, http://www.uclouvain.be
- ISTI, http://hiis.isti.cnr.it
- SAP AG, http://www.sap.com
- GEIE ERCIM, http://www.ercim.eu
- W4, http://w4global.com
- FUNDACION CTIC http://www.fundacionctic.org

Glossary

- **ECA / Embodied Conversational Agent**: A User Interface that graphically aims to unite gesture, facial expression and speech to enable face-to-face communication with users, providing a powerful means of human-computer interaction.
- **Key Performance Indicator (KPI):** A metric used alone, or in combination with other KPIs, to monitor how well a business is achieving quantifiable objectives. In the SAP UCD methodology, a composite "usability" KPI consists of measures of user effectiveness, user efficiency, and user satisfaction.
- **Wireframe:** A wireframe, also known as a schematic or screen blueprint, is a visual guide that represents the skeletal framework of a User Interface (UI). The wireframe depicts the UI layout or arrangement of the website's content, including interface elements and navigational systems, and how they work together
- **WIMP:** A style of interaction using the elements "window, icon, menu, pointing device".
- **RFID:** A technology that uses radio waves to transfer data from an electronic tag, called RFID tag or label, attached to an object, through a reader for the purpose of identifying and tracking the object.
- **NASA-TLX:** A subjective, multidimensional assessment tool that rates perceived workload on six different subscales: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration.
- **HMD**: A head-mounted display or helmet mounted display, both abbreviated HMD, is a display device, worn on the head or as part of a helmet, that has a small display optic in front of one (monocular HMD) or each eye (binocular HMD).
- **CRUD**: In computer programming, Create, Read, Update and Delete (CRUD) are the four basic functions of persistent storage
- http://www.serenoa-fp7.eu/glossary-of-terms/