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

Project no. FP7 – ICT – 258030

Deliverable D5.2.2 Application Prototypes (R1)

Due date of deliverable: 31/08/2012

Actual submission to EC date: 31/08/2011

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	ISTI
Approved by	Telefónica I+D
Project Officer	Michel Lacroix

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

Changes			
Version	**Date**	**Author**	**Comments**
0.1	14/05/2012	SAP	First draft send to reviewer
0.2	18/05/2012	CNR-ISTI	Reviewed version send to SAP
0.3	21/05/2012	SAP	Reviewed draft with structure and contributors sent to partners
0.4	08/06/2012	TID, W4	Partner content input
0.5	15/06/2012	SAP	Consolidated version
0.6	02/07/2012	CNR-ISTI	Reviewed version send to SAP
0.7	06/07/2012	SAP, TID, W4	Final version send to reviewer
1.0	17/08/2012	TID	Final version send to EC

Executive Summary

This deliverable presents the second release of the Serenoa application prototypes with a focus on design and 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 Transaction scenario of W4 and the Warehouse Management scenario of SAP are three independent scenarios, for which the prototypes are being developed. The agile methodology describes how the software development life cycle (SDLC) of the prototypes can be supported. The focus of this description will be on the gathering, designing, implementation and evaluation of use cases and adaptation rules. Making use of the Serenoa framework technical solutions, the prototypes are continuously gaining fidelity and functionality. The current state of the prototypes, their requirements and the implemented or planned 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' will investigate three prototype applications based on technologies developed in the previous work packages.
- Task T5.3 'Evaluation' will evaluate the functionality, usability, effectiveness and quality of the developed technologies from a practical perspective.

In this second phase of task T5.2 'Prototype Development' the focus is on requirements, development of the application prototypes, as well as the integration of the Serenoa framework developed throughout different work packages into the current prototypes. The previously gathered requirements were revisited in the light of the recently expressed feedback from the EU reviewers. The descriptions of the prototypes were enhanced by including adaptation use cases and the planned or realized integration of the Serenoa use cases. A description of the agile methodology was included, aiming at supporting software development life cycle (SDLC) of adaptive SFEs. For this, best practices and successful patterns from exiting methodologies, like Scrum and User Centered Design (UCD) have been identified and adjusted to the particular aspect of adaptation.

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 5.2.1 is the first 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 5.1.1 Serenoa Framework which describes the technical modules developed within the Serenoa project
- Deliverable D3.4.1 Agile Methodology Description (R1) presents the first release of the Serenoa agile methodology description.
- D3.1.1 "Reference Model Specification" which 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. An actual and updated list of requirements and validation criteria for the prototypes is presented in section 3. The agile methodology is described in section 4. It summarizes the methodologies Scrum and UCD and shows their relevance for the Serenoa prototypes. Common patterns are suggested which have been identified as useful for the adaptive characteristic of the prototypes. The main contribution of this deliverable is presented in the section 5, 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 6 summarizes the contributions of the deliverable and section 7 gives directions for future work.

2 Scenarios

In this chapter we will describe the scenarios of the Serenoa project. Compared to previous deliverables, two scenarios are now described in a more specific and detailed way. As argued in previous deliverables, 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. This flexibility is an essential characteristic of scenarios [2], which can also be seen from common, very broad definitions of scenarios, such as "Scenarios are stories. They are stories about people and their activities." [3]. The purpose of scenarios is to help the developers, customers and any other stakeholders to understand who the end-users are, what (product/ tool/ application) will be developed and how the users will interact or use them.

Each industrial partner in Serenoa has designed an own application scenario upon which the prototypes will be developed. These scenarios were already presented in D5.2.1 but due to the above mentioned changes, they will be presented here again. The original three scenarios were:

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

Whereas in D5.2.1 "Application Prototypes (Requirements and Design)", the first low fidelity prototypes were presented, here we will examine them in more detail and try to provide more concrete examples of usage. The current prototypes emerging from these scenarios and our expectations concerning the user experience will be presented later in chapter 4.

2.1 E-Health Scenario

The scenario for TID's Serenoa prototype is aimed at providing a seamless multi device experience to users of two TID pilot programs in the field of E-Health, whose different approaches will be summarized here:

- The SARA project is intended to provide a user interface for chronic disease patients self-monitoring in the form of a (Windows based) tablet PC. The project is currently evolving to provide multi-device access to the application, via the use of regular Windows desktop computers, Android and iOS tablet devices and smartphones; it is also exploring the possibility of introducing TV-based devices. This project is now in a pre-market phase, after successful field tests using real patients from the Andalusian health system.
- The HealthDrive pilot program aims to leverage on consumer devices such as computers, tablet PCs and phones to provide its users access to their personal file on the Andalusian health system. In order to do so, all medical information is digitized and shared by the institutions, with a publicly accessible interface for each user in which she/he can interact with doctors and see their health records. The Andalusian Health System[1] is the official public health system for the Andalusian region in Spain, providing universal health care to its nearly 8.5 million inhabitants. The system is currently in the process of having its centres and processes completely digitalised to provide a faster and more efficient service to its beneficiaries. Telefónica I+D is one of the major entities providing expertise to the public office in order to advance towards its objectives.

One of the clear advantages of this on-going process will be the lowering of the obstacles for the users to be able to query or administer their patient data. It is envisaged that soon all of the data will be available to users online, therefore easing the management of it and enabling that users may perform more advanced actions such as sharing data with relatives or other doctors outside of the health system.

Another significant driver for progress in these programs would be the significant reductions in the chronic patients' management costs: a on-line self-assessment software such as the one in the Chronic Patients

[1] Consejería de Salud de Andalucía: http://www.juntadeandalucia.es/salud (Spanish only)

scenario will diminish the economic impact on the health system that the tens of thousands of these patients pose. Additionally, patients are also satisfied with these remote management tools because they need to be in touch with the health system in person is also reduced and so their quality of life is improved.

The Serenoa Telefónica I+D group is going to provide these two pilot projects with new concepts of UI based on an interactive avatar in order to augment the possibilities of SFEs. To achieve this goal, TID will leverage their past experience with such systems and improve it with Serenoa concepts and technology to solve the difficulties posed with using the avatar in the wide array of supported devices.

2.2 E-Commerce Transaction Scenario

The goal of 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 two particular aspects will be explored:

- Multimodality (based on the type of device accessing the application),
- Accessibility, which refers mainly to the practice of making web sites usable by people of all abilities and disabilities.

In this scenario, we will not deal with disabilities such as blindness or deafness, which require more sophisticated hardware devices and specific users' profiles for testing. Rather, we will illustrate how the application content and operations can be made understandable, with predictable behaviours, by different profiles of users.

The scenario will be defined by a business process with typical exchanges between a front office application (for customers) and a back office application (for the seller). To become a customer of this service, customers create an account and enter, once for all, their payment details. Afterwards, to make a payment, they authenticate by simply using their username and password. The company from which this use case is inspired earns revenues from keeping a fraction of the transaction amount and serves as an intermediary between the product providers and the customers. The providers are not paid before the items are effectively shipped and received by the customer. This secured end-to-end service is a way of increasing confidence on both sides of the transaction. Because the goal of Serenoa is to illustrate how SFEs can adapt to the user's context, and not to explore E-Commerce solutions, we will make the following simplifying assumptions:

- We will deploy the features using 3 applications to which the various actors connect to play their role, as defined in the business process:
    - Two different front-office applications, for customers to shop, authenticate order and pay for their items: one deployed as a web application, the other as a mobile application. For mobile device, target platforms.
        - The first instance is based on HTML5/JQuery (web browser access only, but offers support for disabilities like color blinded)
        - The second instance illustrates portability across platforms (Web browser, mobile terminals with Andoid and IOS players)
    - One back-office application for the use of employees to follow-up customers' requests, from order to item delivery. Adaptation to user preferences (language, colors, fonts, skins, hide/show/order/filter information from pages, possibility to demonstrate cross platform capabilities with rich swing client)
- A simplified item catalog will be used for the purpose of the demo. We will use an online bicycle shop selling bikes, accessories and clothing.
- Back-office users will be registered in some data system with their specificities.
- The intent is not to offer a comprehensive workflow by exploring all the branches of the business process and manage all the cases that may result in a failed transaction (for example if items are not available or if their timely delivery fails).

This scenario will illustrate the differences in the application's behavior depending on 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).

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. Here workers run off storage racks in a warehouse intending to pick the necessary parts. The workers of the succeeding assembly step strongly rely on a correct consignment. An uncompleted consignment will lead to a downtime of the production line, which is translated into losses for the company. In a similar manner, inaccurate pickings will decrease the quality and increase workers' frustration. Therefore, the picking process makes high quality and time demands on the workers. Especially the fact that warehouses employ unskilled workers to relieve the workload during peak times makes consignment a critical task and bottleneck in the overall process. The problem is that workers are often unfamiliar with warehouse settings.

A possible scenario for such a warehouse picking process is described in the following section. As compared to the Warehouse-Management Scenario from D1.1.1, the scenario has been updated. While the setting for the scenario remains the same (e.g. picking process in a large warehouse), the process of picking and the technological support by a multi-device adaptive system is described in more detail:

Ludovic: Ludovic is 40 years old, born in Regensdorf, Switzerland. He lives with his wife and son in Wangen. For the last 3 years he's been working as a picker at the central warehouse of a large wholesale. He works 5 days a week from 7:00 to 16:00. After work he goes to take his son home from kindergarten. He loves spending the weekend with his family.

At work: During his shift Ludovic completes around 30 orders. An order consists of around 40 picks. Ludovic goes around the warehouse with a small electric vehicle with two containers attached to the back to put the articles in. Quite often those containers are destined for different supermarkets and therefore it is important that Ludovic doesn't mix them up.

Every morning…: Every morning when he arrives at work Ludovic goes to his locker and changes into his work clothes. He then picks the items of his "helping system", which can be described in 3 modules:

- **A Voice System** which comprises of a headset and a small device worn on the belt. This system gives and receives verbal commands.
- **Two Scanner Units**, which are small mobile devices capable of scanning RFID and Barcode information.
- **A Head-Mounted Display (HMD)** worn on the head together with the Voice System, which shows additional information on Ludovic's orders.

Today, the story is:

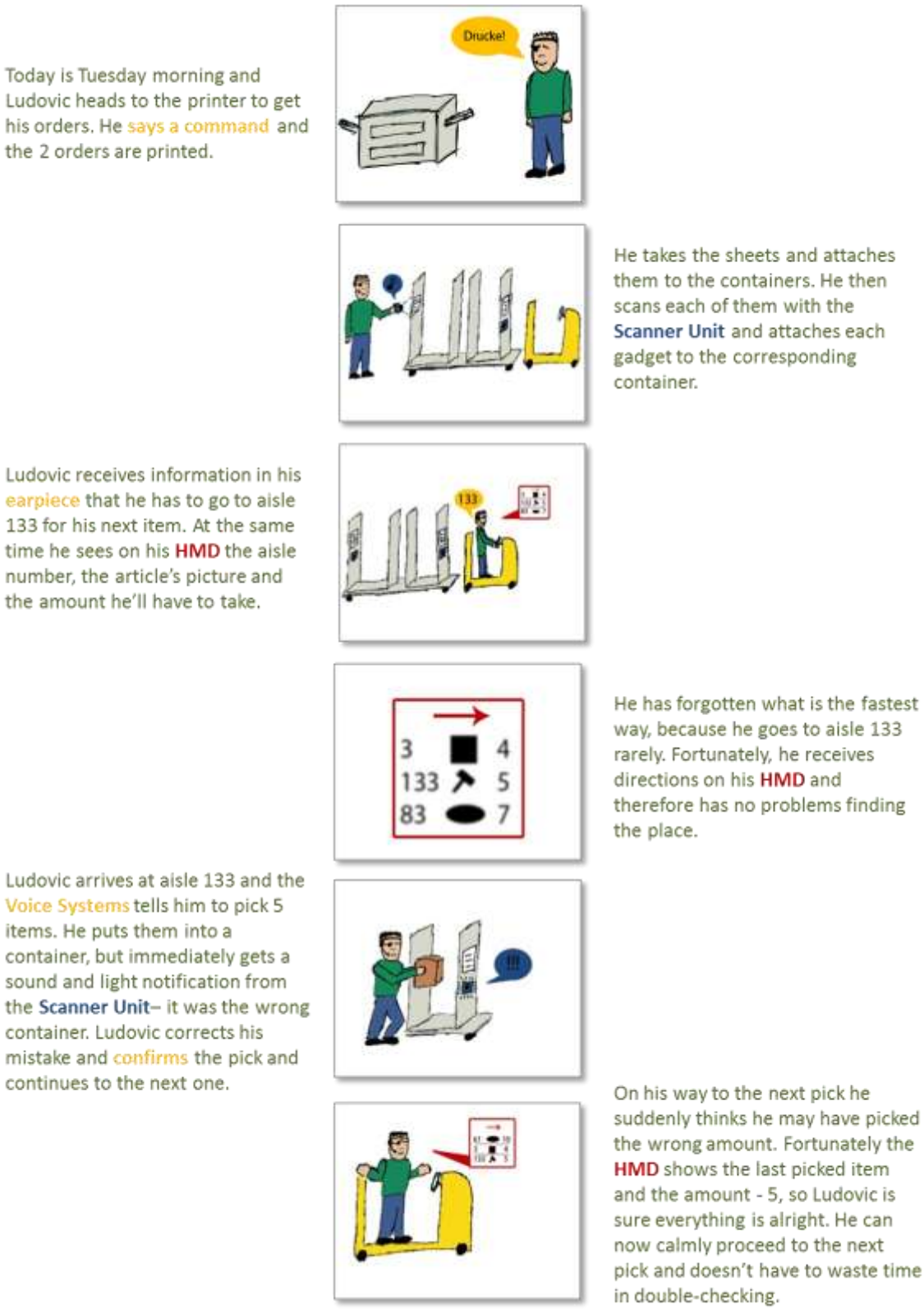

Figure 1: Warehouse-Management Scenario

For temporary workers (novice), who aren't familiar with the warehouse (location) and the mode of operation, the system adaptively gives additional support: more detailed navigational aids, explanations for particular working areas and additional information about the product (e.g. fragility which is important for

stacking the goods). The system also supports different languages. The language is stored in the user's profile, so the correct language is automatically chosen after login to the system.

3 Requirements and validation criteria

This chapter presents the requirements and validation criteria for each scenario's technical solution. The requirements and validation criteria are taken from the deliverable D.1.1.2. Compared to D1.1.2, some changes were applied in order to continuously enhance the quality of the requirements list. Especially, a table format summarizing all information associated with a particular requirement was applied in order to have a better overview on the huge amount of requirements and associated evaluation criteria (see Table 1). We did not find any established or standardized table format appropriate for our use. The search for an appropriate format in different standards for software requirements (e.g. IEEE 830 and ISO/IEC 25010) did not reveal an applicable format. The format we present here is partially based on the VOLERE Requirements Specification Guideline [Robertson 2006], however it has been adapted to better suit our needs (e.g. the "current state" was added and some inapplicable units like "dependencies" were removed). Further changes as compared to the deliverable D1.1.2 are:

- Some requirements were re-formulated and/or described in a more detailed manner, in order to continuously enhance the quality and concreteness of the requirements.
- A running index to uniquely identify the requirements across the scenarios was added
- The priority of the requirements was added to help the prototype development teams in setting the right focus.

Number	Unique Number of Requirement.
Name	Short name of requirement.
Detail Definition	Detailed information on requirement.
Validation Criteria	The criteria to be verified, i.e. specific criteria which must be met to consider the requirement as fulfilled. "*Inspection*" means that it will not be evaluated in a user study, but verified by pure inspection of the development team. "*To be determined*" means that the exact method of evaluation and therefore the validation criterion is not yet determined.
Priority	M = Must, S = Should, C = Could
Current State	Current state of implementation. Five possibilities: • implemented • partly implemented • to be implemented • not to be implemented • not applicable (for requirements which cannot be implemented directly, like e.g. "satisfaction").

Table 1: New format for requirements list

3.1 Requirements and validation criteria from E-Health scenario

Number	1.1
Name	Audio Feedback
Detail Definition	The system will provide Text-to-Speech but also audio feedback associated to non-verbal events.
Validation Criteria	Inspection. The System must provide some audio feedback (e.g. text-to-speech).
Priority	S

Current State	Implemented

Number	1.2
Name	Visual Feedback
Detail Definition	The system will provide visual feedback for conveying information (e.g. supporting the turn-taking). Some of the following means could be considered: • Non-verbal (visual) communication from the avatar. • Focusing the viewport on a part of the avatar's body. • The use of lightning in the scene.
Validation Criteria	Inspection. The system must provide at least one of the aforementioned visual cues.
Priority	M
Current State	Implemented

Number	1.3
Name	Multimodality
Detail Definition	It consists on the graceful combination of several modalities. For example, the coordination of visual data (e.g. medical graphs) with the avatar behaviour (e.g. gestures or voice).
Validation Criteria	Inspection. The system needs to support at least one modality other than visual output/manual input, e.g. voice in-/output or gesture.
Priority	C
Current State	Partly implemented

Number	1.4
Name	Customization
Detail Definition	The system will provide avatar-oriented customization (different physical aspect, synthesized speech qualities, gender, avatar clothing, gesture repertoire, etc.)
Validation Criteria	A questionnaire will be used for evaluation. Users will be asked if the avatar customization supported by the system is enough.
Priority	M
Current State	Partly implemented

Number	1.5
Name	Control over the adaptation process
Detail Definition	The system will provide the possibility to dispose of the avatar completely and/or use just the speech version without a physical visualization of the character.
Validation Criteria	Inspection: the requirement is planned to be evaluated by checking if the user always has the possibility to select one of the aforementioned possibilities (avatar-based and/or voice-based).
Priority	M
Current State	To be implemented

Number	1.6
Name	Cross-Platform Consistency
Detail Definition	Although it won't be possible to keep the avatar characteristics in devices across the board, extra effort will be put in delivering a consistent avatar, even if the functionality is compromised in some cases. As an example, videos of the avatar may be used, but even in those videos the look and feel and behaviour of the avatar should be kept consistent.
Validation Criteria	A questionnaire will be used for evaluation. Users' perception about the avatar's identity (i.e. look-and-feel, behaviour, etc.) running on different devices will be requested.
Priority	S
Current State	Partly implemented

Number	1.7
Name	Satisfaction
Detail Definition	Users should be satisfied with the overall system.
Validation Criteria	A questionnaire will be used for evaluation. Some of the factors involved in the user's satisfactions are already defined in D2.4.1 [Serenoa Project D2.4.1, 2011] and some others, related to Embodied Conversational Agents, need to be defined [Mencia et al. 2008].
Priority	M
Current State	Not applicable

Number	1.8
Name	Search
Detail Definition	The system will offer search capabilities to patients in order to browse health-related issues or personal information.
Validation Criteria	Inspection: the e-health application provides a way to make searches.
Priority	C
Current State	To be implemented

Number	1.9
Name	User-dependent Adaptation
Detail Definition	The system, including the avatar, will adapt automatically to the needs of different users.
Validation Criteria	Inspection: It will be seen as fulfilled when the system adapts to at least one user-characteristic (e.g. language, experience, preferences, etc.)
Priority	C
Current State	To be implemented

Number	1.10
Name	Anticipation of Events
Detail Definition	The system will anticipate events (as the prototype will need to track situations so it can anticipate issues that may trigger adaptation). For example conflicts into the pill planner schedule or offering direct access to functionalities that are more likely to be used at a particular moment (i.e. take blood pressure).
Validation Criteria	Inspection: the system must be able to anticipate some user-related events and take actions accordingly.
Priority	C
Current State	To be implemented

Number	1.11
Name	Accessibility
Detail Definition	The system needs to offer functionalities to keep the prototype accessible for users with some of the following impairments: • Users speaking different languages than the

	default language. • Motor disabilities. • Cognitive disabilities. • Elderly people.
Validation Criteria	Inspection: the system incorporates means to make accessible the service to at least one of the abovementioned users groups.
Priority	M
Current State	Partly implemented

Number	1.12
Name	System and Task Continuity
Detail Definition	The application should be able to engage users in their daily tasks. Notice E-Health scenario, among other things, is intended to guide the patient through not very amusing tasks as attending medical appointments or taking regularly medication. The avatar will be responsible to instruct patients on these tasks and be persuasive along the sessions.
Validation Criteria	A questionnaire will be used for evaluation. Patients will be asked about the commitment to follow the avatar's advices, how convincing was the avatar, etc. [Berry et al., 2005]
Priority	S
Current State	Partly implemented

Number	1.13
Name	Dynamic User Choice
Detail Definition	Dynamic User Choice, as the users should be able to select their preferred UI configurations (see requirement 'Customization') *easily* for the system, crucially including aspects relating to the avatar presence and/or actions.
Validation Criteria	A questionnaire will be used for evaluation. Users will be asked if they have found easily the configuration options.
Priority	S
Current State	Partly implemented

Number	1.14
Name	Intuitiveness

Detail Definition	The interaction with the system should be easy to learn from the beginning for novice users yet this ease should not be a burden to more advanced users.
Validation Criteria	A questionnaire will be used for evaluation. The interface must be rated 'sufficiently' about its naturalness and easiness of use.
Priority	M
Current State	Implemented

Number	1.15
Name	User Trust
Detail Definition	Bearing in mind the application field (e-health), the system needs to be trusted by users so they take its advice into account as a medical tool. However, special care has to be taken into consideration so that users don't fool themselves into thinking that this is a system continually backed and monitored by medical staff.
Validation Criteria	A questionnaire will be used for evaluation. The system must be at least rated as "sufficiently trustworthy".
Priority	S
Current State	Partly implemented

Number	1.16
Name	Confidentiality and Integrity of Data
Detail Definition	The system uses sensitive data (medical reports, images, etc.), so the data should be kept secure and foremost, the system needs to be perceived by patients as it was secure.
Validation Criteria	A questionnaire will be used for evaluation. The system must be at least rated as "sufficiently secure".
Priority	S
Current State	Not applicable

3.2 Requirements and validation criteria from the E-Commerce transaction scenario

Number	2.1
Name	Easy Connection and Configuration
Detail Definition	The system will enable the end users to configure their environment easily.
Validation Criteria	Inspection: all available platforms need a fast and easy way to login to the system

	and access the settings menu.
Priority	M
Current State	Implemented for connection, partly implemented for configuration.

Number	2.2
Name	Cross-Platform Consistency
Detail Definition	The application will run consistently across different platforms.
Validation Criteria	Inspection: no major differences may be shown for: • The interface • The interaction style • The functionalities offered
Priority	M
Current State	Partly implemented: almost finished for web based and rich client platform, partly for mobile players Android and iOS.

Number	2.3
Name	Platform-dependent Adaption
Detail Definition	The application will run on mobile devices.
Validation Criteria	The application will run at least for one of the following mobile operating systems: • iOS • Android
Priority	M
Current State	Implemented

Number	2.6
Name	Accessibility
Detail Definition	The system will be accessible for at least one specific group of users.
Validation Criteria	Inspection: the system needs to offer functionalities to keep the prototype accessible for users with at least one of the following impairments: • users speaking different languages than the default language • motor disabilities • cognitive disabilities • elderly people
Priority	S
Current State	Partly Implemented

3.3 Requirements and validation criteria from the Warehouse Management scenario

Number	3.1
Name	Stability
Detail Definition	The system needs to be stable, so that no interruptions will occur during its runtime.
Validation Criteria	Inspection: the system must run continuously and without interruptions for a test period of one hour.
Priority	S
Current State	To be implemented

Number	3.2
Name	Error
Detail Definition	The system will show the warehouse clerks when a mistake was made and how they can correct it.
Validation Criteria	The requirement will be seen as fulfilled, when the subjectively perceived error rate with the current system is below the subjectively perceived error rate with the old system.
Priority	M
Current State	Implemented

Number	3.3
Name	Response Time
Detail Definition	The run-time algorithms of the system will not stop the user from interacting in a fast way.
Validation Criteria	Not to be evaluated, since slow response time is assumed to be reflected in other metrics, such as performance, efficiency and satisfaction.
Priority	S
Current State	To be implemented

Number	3.4
Name	Performance & Efficiency
Detail Definition	The system will allow the user to perform fast and efficient without making mistakes.
Validation Criteria	The requirement will be seen as fulfilled, when the subjectively perceived performance and efficiency with the current system is below the subjectively perceived performance and efficiency with the old system (i.e. a voice-only system without graphical UI on the HMD).
Priority	M

Current State	Implemented

Number	3.5
Name	Satisfaction
Detail Definition	Users should be satisfied with the overall system.
Validation Criteria	A questionnaire will be used for evaluation. Users must be at least “satisfied” with the system.
Priority	M
Current State	Not applicable.

Number	3.6
Name	Intuitiveness
Detail Definition	The interaction with the system will be intuitive.
Validation Criteria	A questionnaire will be used for evaluation. The system must be at least rated as “sufficiently intuitive”.
Priority	S
Current State	Not applicable

Number	3.7
Name	Simplicity
Detail Definition	The system will be designed in a simple (easy to use) way.
Validation Criteria	Not to be evaluated, since lacking simplicity is assumed to be reflected in other metrics, such as performance, efficiency and satisfaction.
Priority	S
Current State	To be implemented

Number	3.8
Name	Learnability
Detail Definition	The system will be easy to learn.
Validation Criteria	A questionnaire will be used for evaluation. Users must be at least “satisfied” with the learnability of the system.
Priority	S
Current State	Not applicable

Number	3.9
Name	Identification
Detail Definition	The user will need an identification code to access the system.

Validation Criteria	Inspection: The requirement will be seen as fulfilled, when the user needs an identification code to start the system.
Priority	S
Current State	To be implemented

Number	3.10
Name	Easy Connection and Configuration
Detail Definition	It must be easy for the user to connect to the system and to configure it.
Validation Criteria	A questionnaire will be used for evaluation. Users must be at least “satisfied” with this aspect of the system.
Priority	1
Current State	To be implemented

Number	3.11
Name	Working Environment
Detail Definition	The system will be adaptive to the working environment.
Validation Criteria	The requirement will be seen as fulfilled when the system adapts to at least one aspect of the environment (e.g. light or sound conditions, location of the worker).
Priority	M
Current State	Implemented

Number	3.12
Name	Independence of Different Technological Spaces
Detail Definition	The system will be adaptive independently of the different technological spaces.
Validation Criteria	The requirement will be regarded as fulfilled, when the system works at least with two different devices, e.g. the HMD and a tablet PC or a smartphone.
Priority	S
Current State	Implemented

Number	3.13
Name	Cross-Platform Consistency
Detail Definition	The application will run consistently across different platforms.
Validation Criteria	Inspection: no major differences may be shown for: • The interface • The interaction style • The functionalities offered
Priority	C

Current State	Not applicable

Number	3.14
Name	System and Task Continuity
Detail Definition	The adaptation should run in a continuous process.
Validation Criteria	Inspection: The continuity will be regarded as fulfilled when no interruptions attributable to the adaptation can be observed.
Priority	S
Current State	Partly implemented

Number	3.15
Name	Control Over the Adaptation Process
Detail Definition	The user will have the control over the adaptation process.
Validation Criteria	Inspection: the requirement is planned to be evaluated by checking if the user always has the possibility to influence the adaptation. It will be considered as fulfilled when there is at least one possible way to manipulate the adaptation (e.g. stop, change or start it).
Priority	C
Current State	Partly implemented

Number	3.16
Name	User-dependent Adaption
Detail Definition	The system will adapt automatically to the user.
Validation Criteria	Inspection: It will be seen as fulfilled when the system adapts to at least one user-characteristic (e.g. language, experience, preferences).
Priority	S
Current State	Partly implemented

Number	3.17
Name	Multimodality
Detail Definition	The system will provide multimodality.
Validation Criteria	Inspection. The system needs to support at least one modality other than visual output/manual input, e.g. voice in-/output or gesture.
Priority	S
Current State	Implemented

Number	3.18

Name	Visual Feedback
Detail Definition	The system should give immediate visual feedback about task-relevant information.
Validation Criteria	Inspection: The system needs to update the graphical UI in real time to the user's activities.
Priority	S
Current State	Implemented

Number	3.19
Name	Accessibility
Detail Definition	The system will be accessible for at least one specific group of users.
Validation Criteria	Inspection: the system needs to offer functionalities to keep the prototype accessible for users with at least one of the following impairments: • users speaking different languages than the default language • motor disabilities • cognitive disabilities • elderly people
Priority	C
Current State	To be implemented

Number	3.20
Name	Customization & Personalization
Detail Definition	The system will provide the possibility to the user to customize/personalize it by changing the settings.
Validation Criteria	Inspection: by asking the question: "Does the system provide the means (e.g. a settings menu or wizard) in which the user can actively adapt aspects of the system/interface?" For the requirement to be fulfilled, this question has to be answered positively.
Priority	S
Current State	To be implemented

4 Agile Methodology Description

The previous chapters reflect the effort of defining and updating the scenarios and gathering the requirements for the prototypes. In comparison to earlier versions of this and other documents it can be seen that both, scenarios and requirements are subject to a continuous evolution resulting in a change. Concerning the requirements the question arises if it is feasible to implement all of them at once concluded with a final evaluation. This chapter proposes an approach which might be better suited to deal with changes and short development cycles.

The aim of Serenoa is to provide a framework which allows developers to create UIs which will be adapted to the context of use, such as:a person's devices, tasks, preferences, and abilities. These adaptive UIs intend to improve the people's satisfaction and performance compared to traditional SFEs based on manually designed UIs.

This document shows the development of adaptive UIs, referred to as application prototypes, through requirements, scenarios, adaptation use cases, design prototypes and technical realization. This section argues that the software development life cycle (SDLC) for adaptive UIs should be agile and user centred. Thus, this section discusses the fusion of these methodologies applied mainly to the gathering, design and implementation of adaptation rules. We do not aim at defining a brand new agile methodology. Instead, we aim at identifying best practices and successful patterns already present in methodologies such as Scrum that are particularly important for context-aware applications and at adjusting them to serve Serenoa's goals more efficiently. We believe that reusing some tools and concepts emphasized by User Centred Design (UCD) can be beneficial to the development of SFEs. The combination of such ideas, although already researched by other projects, was experienced when implementing the different Serenoa prototypes.

The development team needs to get a clear idea about which are the specific needs of adaptation for the end-user. These needs must be collected and formalized as adaptation rules. It is clear that some kind of waterfall model, i.e. first collecting all adaption rules, then fully designing the UI and finally implementing the software, has a high risk of not fitting to the user's needs.

Agile software development is becoming increasingly popular in companies and organizations. This approach, when combined with UCD, incorporates some regular feedback on the current piece of software from users in order to respond to changes. In the case of Scrum regular demonstrations of the product to stakeholders are expected at the end of every Sprint. The process of putting the users' needs in the focus of the development has been formalized in the methodology of UCD by defining cyclic events (Research, Design, Adapt, and Measure).

This document does not intend to cover all aspects of the development of the Serenoa application prototypes. Some of these aspects are quite general and may apply to any software development. Instead we have chosen to concentrate on a specific feature which is particular for the Serenoa application prototypes: The adaptation rules. The adaptation rules seem to be quite suitable to be placed in the focus of an agile methodology. The reason is that it seems quite unrealistic that all adaptation rules can be gathered at once through a common requirements engineering approach. Some adaptation rules might only emerge after a first set of adaptation rules has been implemented and presented. Who can decide which will be the primary adaptation rules? Who can evaluate the rules? We believe that the answer to this lies in an agile methodology fused with aspects of UCD.

In [6] the role of Interaction Design which is integrated in an agile process is seen as

- Front-end UI development alongside developers
- Work with Product Owner to articulate and validate the product concept

As patterns for success [6] describes

- Put UX in the navigator's seat (as part of the Product Owner team)
- Research, model, and design upfront, but lightly
- Plan for continual ongoing user contact for research and validation
- Focus on the big picture

- Use parallel track development to work ahead, and follow behind

Similar integration points have been stated by [7]:

- Pre-Sprint Iteration of Design
- Defining a small section of the requirements, verifying and designing these requirements, implementing the resulting features, testing the implementation, and then beginning again.
- Regular feedback from users
- Retaining the context of use to ensuring a holistic rather than a fragmented and disjoined approach to the solution

In [8] the authors proposed a new approach to improve Software Development by applying User Centered and Model-Driven Development in an Agile manner but no adaptiveness of the UI was investigated in that study.

4.1 Scrum

As already mentioned in a previous deliverable [10], Scrum is centred on the persons; the customer must have all the needs satisfied. Its most important characteristic is the fact that changes in the requirements are considered in a dynamic way. Thus the customer is allowed to test and provide feedback for the project as soon as possible. This method is also iterative and incremental, and it has short development cycles. This method is very dynamic and allows quick reactions for the changes in the requirements. In such a way, it is recommended for highly dynamic environments characterized by fast changing requirements.

Research projects in general and Serenoa particularly are characterized by fast changing requirements where development teams are required to build complex software. From this point of view, Scrum is a suitable agile methodology for Serenoa. Results on the application of Scrum for the development of the Authoring Tool have been reported in [10]. On the other hand research projects are usually lacking an important stakeholder for the Scrum methodology: the customer. Due to this fact the Product Owner of a research project plays an important role. The Product Owner needs to be empathic enough to foresee the needs of future customers.

Some important entities taken from Schwaber and Sutherland [4] are summarized in the following sections.

4.1.1 The Product Owner

The Product Owner is responsible for maximizing the value of the product and the work of the Development Team. How this is done may vary widely across organizations, Scrum Teams, and individuals. The Product Owner is the sole person responsible for managing the Product Backlog as shown in the left part of Figure 2.

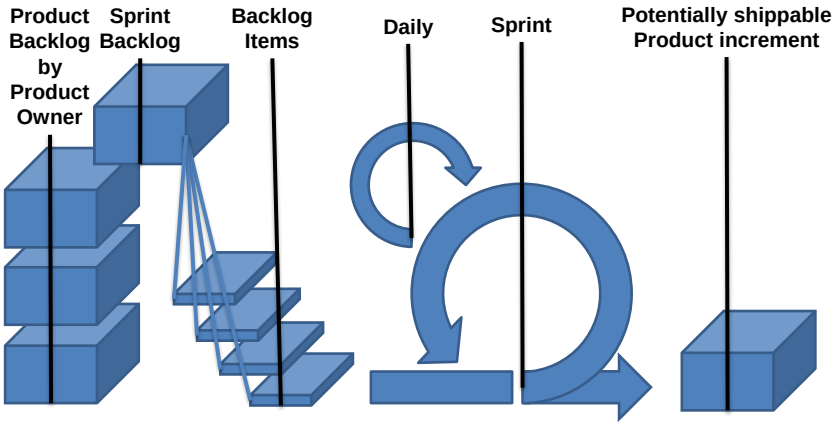

Figure 2: Important entities of Scrum taken from Schwaber and Sutherland [**4**]

Product Backlog management includes:

- Clearly expressing Product Backlog Items;
- Ordering the items in the Product Backlog to best achieve goals and missions;

- Ensuring the value of the work the Development Team performs;
- Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,
- Ensuring the Development Team understands items in the Product Backlog to the level needed.

The Product Owner may do the above work, or have the Development Team do it. However, the Product Owner remains accountable. The Product Owner is one person, not a committee. The Product Owner may represent the desires of a committee in the Product Backlog, but those wanting to change a backlog item's priority must convince the Product Owner. For the Product Owner to succeed, the entire organization must respect his or her decisions. The Product Owner's decisions are visible in the content and prioritization of the Product Backlog. No one is allowed to tell the Development Team to work from a different set of priorities, and the Development Team isn't allowed to act on what anyone else says [4].

In the context of research on the Serenoa application prototypes, described in this document (short: in the context of Serenoa) the Product Owner would need to:

- Clearly express the adaptation rules using Product Backlog Items
- Order the adaptation rules in the Product Backlog
- Ensure the Development Team understands the adaptation rules in the Product Backlog

Those who want to change the adaptation rules' priority need to convince the Product Owner.

4.1.2 The Development Team

The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of "Done" product at the end of each Sprint. Only members of the Development Team create the Increment. Development Teams are structured and empowered by the organization to organize and manage their own work. The resulting synergy optimizes the Development Team's overall efficiency and effectiveness. Development Teams have the following characteristics [4]:

- They are self-organizing. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality;
- Development Teams are cross-functional, with all of the skills as a team necessary to create a product Increment;
- Scrum recognizes no titles for Development Team members other than Developer, regardless of the work being performed by the person; there are no exceptions to this rule;
- Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole; and,
- Development Teams do not contain sub-teams dedicated to particular domains like testing or business analysis.

In the Serenoa context the Development Team does the work of delivering potentially releasable increments of the adaptive UI at the end of each Sprint. The Development Team is cross-functional with experts from user experience to gather adaptation rules and to design mock-ups and UI developers.

4.1.3 The Sprint

The heart of Scrum is a Sprint shown in the centre of Figure 2, a time-box of one month or less during which a Done, useable, and potentially releasable product Increment is created. Sprints have consistent durations throughout a development effort. A new Sprint starts immediately after the conclusion of the previous Sprint.

Sprints contain and consist of the Sprint Planning Meeting, Daily Scrums, the development work, the Sprint Review Meeting, and the Sprint Retrospective.

During the Sprint:

- No changes are made that would affect the Sprint Goal;
- Development Team composition and quality goals remain constant; and,
- Scope may be clarified and re-negotiated between the Product Owner and Development Team as

more is learned.

Each Sprint may be considered a project with no more than a one-month horizon. Like projects, Sprints are used to accomplish something. Each Sprint has a definition of what is to be built, a design and flexible plan that will guide building it, the work, and the resultant product. Sprints are limited to one calendar month. When a Sprint's horizon is too long the definition of what is being built may change, complexity may rise, and risk may increase. Sprints enable predictability by ensuring inspection and adaptation of progress toward a goal at least every calendar month. Sprints also limit risk to one calendar month of cost [4].

In the Serenoa context during a Sprint potentially releasable increment of the adaptive UI prototype is created.

4.1.4 Product Backlog

The Product Backlog as shown in the left part of Figure 2 is an ordered list of everything that might be needed in the product and is the single source of requirements for any changes to be made to the product. The Product Owner is responsible for the Product Backlog, including its content, availability, and ordering. A Product Backlog is never complete. The earliest development of it only lays out the initially known and best-understood requirements. The Product Backlog evolves as the product and the environment in which it will be used evolves. The Product Backlog is dynamic; it constantly changes to identify what the product needs to be appropriate, competitive, and useful. As long as a product exists, a Product Backlog also exists. The Product Backlog lists all features, functions, requirements, enhancements, and fixes that constitute the changes to be made to the product in future releases. Product Backlog items have the attributes of a description, order, and estimate. The Product Backlog is often ordered by value, risk, priority, and necessity. Top-ordered Product Backlog items drive immediate development activities. The higher the order, the more a Product Backlog item has been considered, and the more consensus exists regarding it and its value. Higher ordered Product Backlog items are clearer and more detailed than lower ordered ones. Requirements never stop changing, so a Product Backlog is a living artefact [4].

In the context of Serenoa the Product Backlog is an ordered list of adaptation rules that might be needed in the adaptive UI prototype. The list of adaptation rules is never complete, it is a living artefact. Top-ordered adaptation rules drive immediate development activities.

4.1.5 Sprint Backlog

The Sprint Backlog as shown in the left part of Figure 2 is the set of Product Backlog items selected for the Sprint plus a plan for delivering the product Increment and realizing the Sprint Goal. The Sprint Backlog is a forecast by the Development Team about what functionality will be in the next Increment and the work needed to deliver that functionality. The Sprint Backlog defines the work the Development Team will perform to turn Product Backlog items into a "Done" Increment. The Sprint Backlog makes visible all of the work that the Development Team identifies as necessary to meet the Sprint Goal. The Sprint Backlog is a plan with enough detail that changes in progress can be understood in the Daily Scrum. The Development Team modifies Sprint Backlog throughout the Sprint, and the Sprint Backlog emerges during the Sprint. This emergence occurs as the Development Team works through the plan and learns more about the work needed to achieve the Sprint Goal. As new work is required, the Development Team adds it to the Sprint Backlog. As work is performed or completed, the estimated remaining work is updated. When elements of the plan are deemed unnecessary, they are removed. Only the Development Team can change its Sprint Backlog during a Sprint. The Sprint Backlog is a highly visible, real-time picture of the work that the Development Team plans to accomplish during the Sprint, and it belongs solely to the Development Team [4].

In the context of Serenoa the Sprint Backlog is the set of adaptation rules selected for the Sprint. The Development Team modifies the adaptation rules throughout the Sprint and the adaptation rules emerge during the Sprint.

4.2 User Centred Design (UCD)

According to Rubin [3] User-Centred Design (UCD) is a flexible approach for software development projects that enables teams to more effectively meet the needs and requirements of end users and customers.

Weissenberger and Fellenz-Thompson [5] describe five distinct phases which organize the collaborative activities: Plan, Research, Design, Adapt, and Measure (see Figure 3).

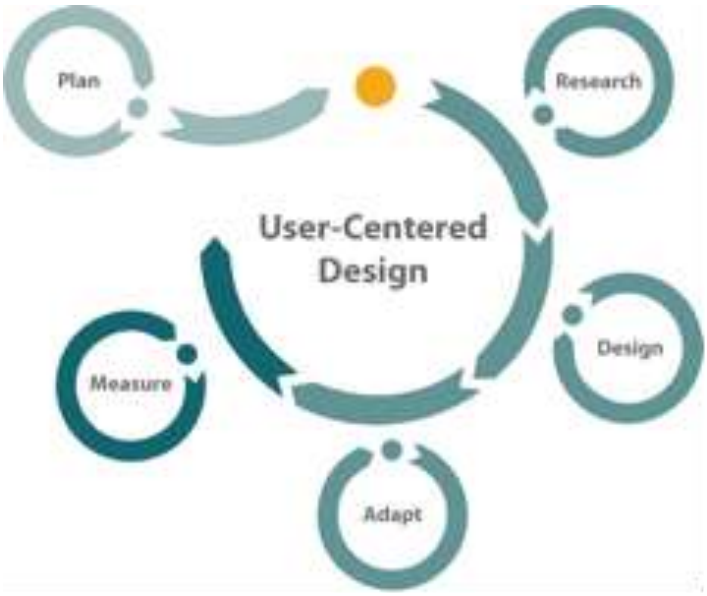

Figure 3: The 5 phases of the User-centered Design approach: Plan, Research, Design, Adapt, and Measure [**5**].

Weissenberger and Fellenz-Thompson [5] suggest that during the Plan phase, the team determines all of the User-Centered Design (UCD) activities and ensures that the necessary resources are available. After planning the (UCD) process Research appears as the second step. The main goal is to understand the needs of the user taking into account goals, tasks, tools and pain points. This can be achieved by interviews and field research. User profiles are created or updated and requirements are extracted by analyzing the data. This step forms a solid foundation for the Design step which takes structures everything that has been learned. In this step, use cases are created and potential UI solutions are sketched. These design proposals are validated with users in the form of formative usability studies or cognitive walk-throughs. In the Adapt step the design is handed off to the development group. As development begins coding the designs will be adapted. This might be due to the discovery of some unforeseen limitations in the target technology, new requirements, or missing functionality in the initial design. The product can finally be evaluated in the Measure step by combing scores for Effectiveness, Efficiency and Satisfaction.

In the context of Serenoa three steps can be used as a basis for the agile methodology as shown in Figure 4. The requirements and user profiles, which have been gathered in the Research step, lead to the writing of Use cases, the sketching of designs and the conduction of tests in the Design step.

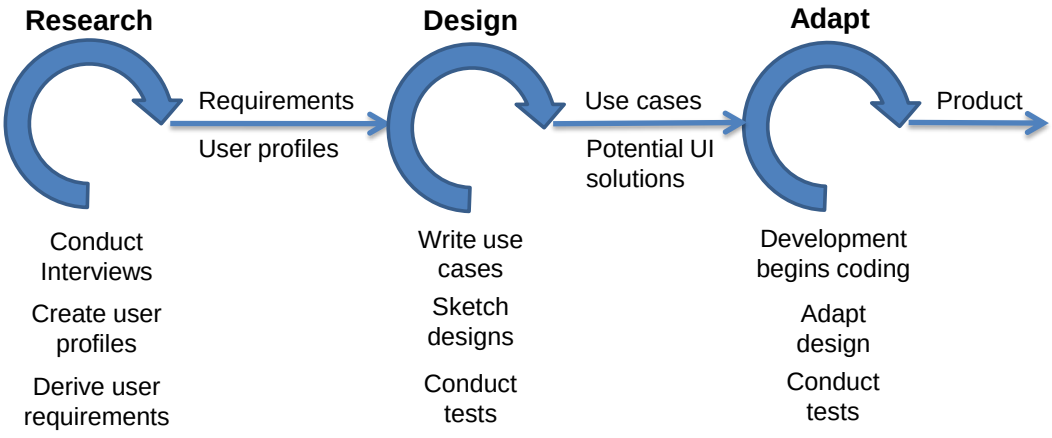

Figure 4: Three phases of the User-centred Design approach with relevance to the Serenoa application prototypes.

In the Serenoa framework, Reference Models (see [11]) that formalize the concepts related to adaptation are also described as Use Cases. The concepts defined by the CARF provide a base to define the Reference Models. These concepts gathered are being continuously refined and updated. For instance, the how branch of CARF lists adaptation techniques, which are also used to define possible use cases of an adaptive or adaptable application. In terms of abstraction layers for the tasks, as defined in the Task and Domain models, they are associated with the use case definitions (i.e., adaptation techniques). In the Serenoa framework Use Case Diagrams defined in the RM module, set potential adaptation techniques that can be performed by end users or by the system.

The outcomes of the Design step are use cases and potential UI solutions which are handed over to the development group. The development group begins coding during the Adapt step which results in a modification of the design. The modified designs are then again validated through the conduction of tests.

The Initial Plan step and the final Measure step are out of scope for this document as the outcome of research projects are usually not at the level of a product which can be sold. The agile methodology described in this document can later be extended for the remaining steps. It might be important to remember that this will also "close the cycle" as shown in Figure 3, that UCD is a cyclic development. This is usually also reflected through different levels of fidelity for the prototypes.

4.3 UCD of adaptive UIs in an agile environment

In the context of Serenoa the fusion of UCD and agile for building adaptive UIs should allow us to build the right adaptation, in the right way. There are some similarities which will help to perform the fusion:

1. Frequent delivery
2. Early delivery
3. Valuing working software over design specification
4. Responding to change.

We now suggest the following actions as parts of the fusion which also regards the differences between UCD and Scrum:

1. One-Sprint Offset between Design and Development
   The Design prototypes like paper-prototypes will be done one Sprint in advance. The Product Owner will use them to shape the Backlog Items for the implementation in the proceeding Sprint.
2. One Sprint Offset between Development and User Feedback
   Each Sprint delivers a potentially shippable product increment, which is also the latest level of design fidelity: working software. During this phase the UI has received a shift of focus from design on interaction to code development which should be verified. Thus, the working software will be used to get feedback from the users.
3. Implementation of a small set of features per Sprint
   Due to action 1 and 2 the design, implementation and verification happens in three Sprints. This means that each 6 – 12 weeks, depending on a Sprint length from 2 – 4 weeks, an evaluated potentially shippable product increment is available. This is only possible if the Scrum Team concentrates on a small set of features and requirements.
4. Continuous verification of the uses cases
   Delivering small increments of the product which are potentially shippable require a continuous verification that these pieces fit together in the bigger picture, i.e. the use case.

In the context of Serenoa the software development process derived from the above actions is:

> *There will be no extensive gathering and description of adaptation rules upfront as a part of a requirements engineering activity. At Sprint n the team members with user experience domain expertise start to gather a first set of adaptation rules from the user and write the corresponding Use Case. The Product Owner will prioritize the adaptation rules and form Backlog Items for getting them implemented in Sprint n+1. In Sprint n+1 one part of the Development Team starts implementing (coding) the first set of adaptation rules while the other part sets out to gather the second set of rules. The Product Owner will then ensure the value of the first potentially shippable*

product increment and prioritize the second set of adaptation rules for implementation in Sprint n+2. In Sprint n+2 the Development Team implements the second set of adaptation rules, gets feedback from the user for the implementation of the first set of rules and gathers the third set of rules.

A first evaluation of the proposed methodology by the Sernoa stakeholders resulted in the following feedback:

- Serenoa's model driven approach allows making Scrum's iterations more effective: when models changes, apps change accordingly at a minimum effort cost.
- Scrum's iterations can sometimes result in a loss of focus: engaging users in a more methodical way can avoid this pitfall.
- Serenoa's focus on the adaptation to the user context makes it natural to put the end-user at the center of the development. The Serenoa prototypes are therefore ideal use-cases to try UCD concepts.

One issue that remains is the task of talking to the end-user. Not all teams have the possibility to talk to those people who will be using their product at the end. In those cases the team should try to do the iterations with people who represent the end-users as good as possible. This might be in some cases simply those people who do not design or code the product.

5 Design of the Prototypes

The following sections divide this chapter into the three application prototypes.

At first, the adaptation use cases are presented, as this is seen to be a very essential feature of the prototypes with regard to the Serenoa project. The uses cases described here are partly taken from the deliverable D3.1.1 “Reference Model Specification” which provides an exhaustive list of possible adaptation use cases which are all reported in the format displayed in Table 2. For this deliverable, we drew on this list to generate a list of adaptation use cases for each single scenario.

Use Case Element	**Description**
Use Case Name	The name of the use case, in a short and clear definition
Use Case Description	A definition of the use case
Primary Actor	Who is the main actor of the use case
Precondition	What are the conditions and requirements to be met before the use case execution
Trigger	What event triggers this use case
Basic Flow	The sequence of steps of the use case when everything is fine (no errors, or exceptions)
Alternate Flows	The most significant alternative flows and exception treatments

Table 2: Use Case Description

The existing or planned 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 systems. An overview on the current framework is given in Figure 5.

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

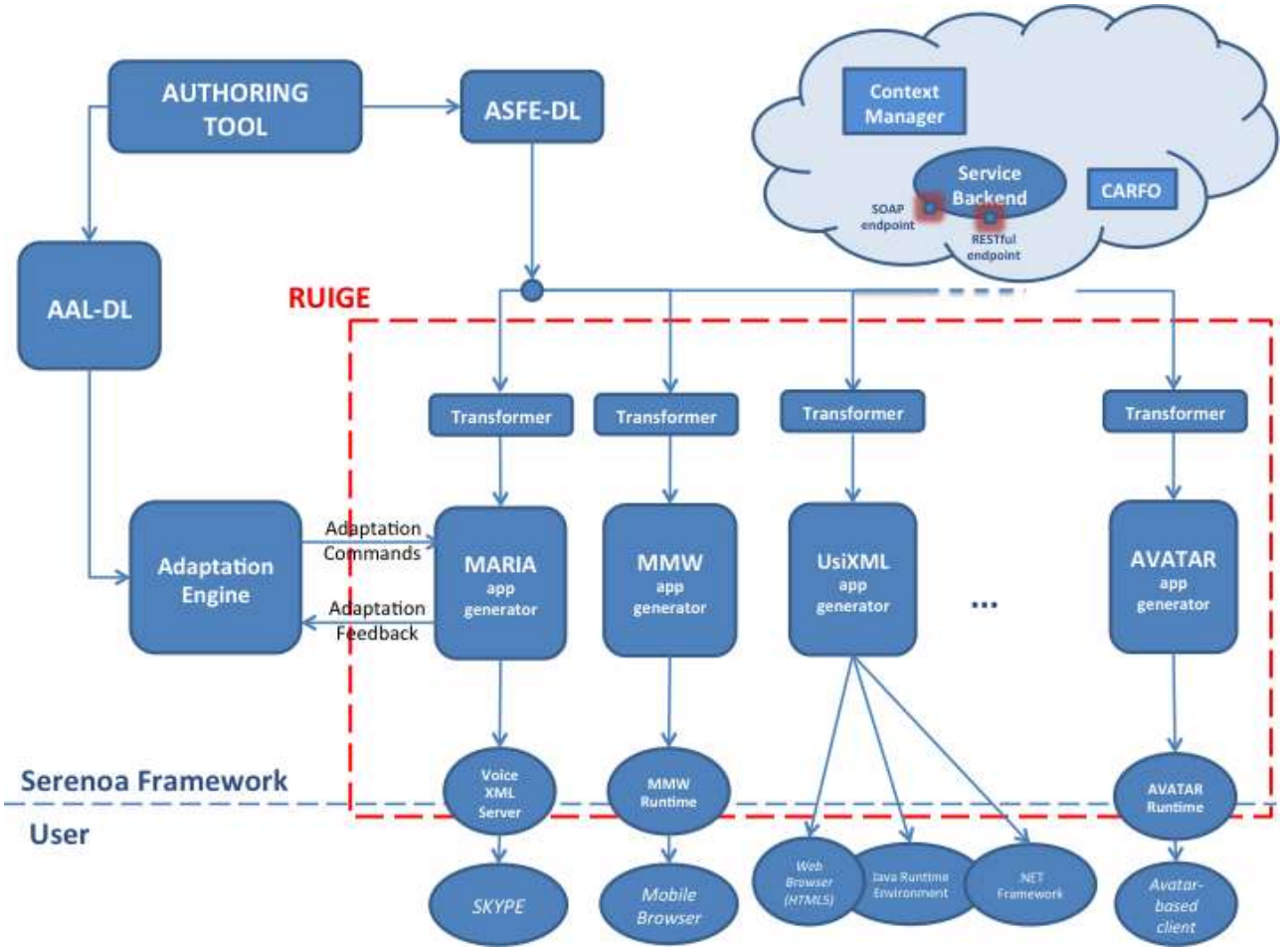

Figure 5: Current Serenoa framework.

5.1 E-Health scenario

The E-Health scenario is aimed at using adaptive ECAs (Embodied Conversational Agents) in order to enrich two different multi-device TID services:

- The SARA project which provides a user interface for chronic disease patients self-monitoring.
- The HealthDrive project which is intended to provide access to the patients' personal file on the Andalusian health system.

5.1.1 Adaptation use cases

In this section, we present several scenarios where adaptation use cases happen. A full compilation of the adaptation use cases is accessible in Annex A-H from D3.1.1 'Reference Models Specifications (R1)', but in this case our aim is to put into the context what are the adaptation strategies and who are the involved stakeholders.

Scenario 1: Jane is American and she is living at Granada. She is pregnant and she is using HealthDrive service to be informed about the progress of her pregnancy. HealthDrive let her to access this personal information in her own language (**Translate Language**). She usually consults her HealthDrive desktop application at home. When she is outside and she tries to access to the eHealth assistant using her smartphone, the avatar presentation is degraded (**Re-size**) and a sequence-of-images version (**Snapshot**) is presented. Afterwards, on the bus, a high level of noise is detected and the avatar voice is no longer audible. The avatar suggests to change the modality (**Suggest**) and Jane agrees (besides she doesn't want the people on the bus were aware of the interaction). Then the volume is turned off (**Change Volume**, **Translate Modality**).

Below further details about the adaptation use cases included in this scenario are presented.

Use Case Name	**Translate Language**
Use Case Description	Given an audio content in a given language, it is translated to another one
Primary Actor	User / System
Precondition	There is audio, and there is a version of the audio content available in another language, or a method implemented that allows the translation
Trigger	The user preferences or the user profile; the location of the user/device.
Basic Flow	There are versions of the content available in different language
Alternate Flows	There is a method available to perform the translation (e.g., a web service)

Use Case Name	**Re-size**
Use Case Description	Modify the size of the content according to the user context (such as preferences, disabilities or constraints of the device)
Primary Actor	User / System
Precondition	There is content (text, UI elements, …) composing the user interface
Trigger	The user has a certain characteristic defined (for instance reduced visual ability)
Basic Flow	The user interface elements are selected, and their sizes are re-defined according to the context (larger or smaller)
Alternate Flows	Besides the re-sizing of the content, it may be also necessary to adjust the layout of the interface

Use Case Name	**Snapshot**
Use Case Description	It consists in taking representative snapshots of a video or an avatar sequence.
Primary Actor	User / System
Precondition	There is an avatar component.
Trigger	The characteristics of the device (e.g. mobile/desktop, power computation, etc.)
Basic Flow	From the avatar behaviour (i.e. gestures, movements, etc.) some snapshots are extracted, trying to be enough representative of the communication act.
Alternate Flows	

Use Case Name	**Suggest**
Use Case Description	Given the context of the user, the system provides also alternative options, such as suggesting additional features, or modalities
Primary Actor	User / System
Precondition	There is content (text, UI elements, …) composing the user interface and also a set of related options that can be suggested to the user during the interaction (or after a task was concluded)
Trigger	This adaptation can be triggered by the accomplishment of a task (for instance, once the user finishes reading an article, the system automatically suggests

	related contents, such as videos, images or texts)
Basic Flow	There is a logic behind associating additional content that is related with the current task of the user, besides there is an event defined to trigger the presentation of the suggestions
Alternate Flows	The suggestions can be provided collaboratively, for instance by users who share the same interests, or are 'contacts' of the end user

Use Case Name	**Change Volume**
Use Case Description	The volume of the sound is modified (to a higher or lower level)
Primary Actor	User / System
Precondition	There is audio content, and its volume can be modified
Trigger	The level of noise in the environment may trigger this technique, the user preference, a user hearing impairment
Basic Flow	The value of a parameter in the player of the audio content is modified accordingly
Alternate Flows	There are different audio contents that can be selected, chosen, according to the volume level

Use Case Name	**Translate Modality**
Use Case Description	Given an audio content it can be converted from audio to text or image for instance
Primary Actor	User / System
Precondition	There is audio content, and alternative versions to provide it in different modalities
Trigger	The user has a hearing impairment that requires alternative versions; the sound system is unavailable or broken; the level of noise in the environment is too high; the user sets its preferences to text (e.g. for privacy concerns, once there is no headset available and she wants to avoid information disclosure)
Basic Flow	There is a version of the content available in an alternative modality
Alternate Flows	There is a method implemented and available to convert the audio content into a different modality

Scenario 2: Maria, an elder woman who has a chronic disease, uses daily SARA interface ('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 (**Alternative Navigation**), although Maria is allowed to hide this component at any time, just by clicking or saying it (**Stretch**). She has some visual problems so the system reads the indications and the graphs or reports which are sent from her doctor (**Translate to Audio**). At the beginning of the interaction the system proposes Maria to measure her blood pressure (**Direct Guidance**) since it has been requested by her doctor. She starts getting her blood pressure but she doesn't remember well how the procedure is. Several errors happen and then the avatar, with a more empathic attitude (**Change of Attitude**), offers her a demo video (**Frame-based**) which shows how it works. This explanatory material is augmented with links to external content (e.g. Wikipedia) in order to clarify the technical terms or the meaning of certain parts of the process (**Annotation of Links**).

Use Case Name	Alternative Navigation
Use Case Description	It consists in designing an application with multiple, and sometimes redundant, ways of navigating.
Primary Actor	User / System
Precondition	There is a navigation scheme in the application that can be replicated
Trigger	There is free space in the screen, the level of expertise of the user, the context of use, the application domain
Basic Flow	To ensure that end users complete their goals, place search and browse navigation tools at the top and start of the page. Position next-step navigation tools toward the top, but opposite the start, as well as at the bottom. Include navigation tools that relate and promote so that users find things that they might otherwise miss, but position these tools further down the page.
Alternate Flows	Allow users to access resources with shortcuts may optimize navigation for expert users

Use Case Name	Stretch
Use Case Description	From the user model and the domain model, the system determines which fragment should be displayed. For each fragment of information there is a short visible place holder. Stretched fragments must be shown and the others shrunk; this decision determines the initial presentation of the fragment, once the content may be stretched or shrunk through mouse clicks for instance
Primary Actor	User / System
Precondition	There is content (text, UI elements, …) composing the user interface, and a user and domain models defining the contents that should be stretched or shrunk
Trigger	A change in the platform type, reducing the screen dimension may trigger this technique
Basic Flow	The user interface elements are selected, and the layout is re-defined
Alternate Flows	

Use Case Name	Translate to Audio
Use Case Description	Given a content, it is translated to an audio format
Primary Actor	User / System
Precondition	There is content, such as a text, or an image, that can be converted into an audio format
Trigger	This adaptation can be triggered when the user is in an environment that unable the interaction in a visual way, e.g. when she is driving a car, or when the user has a visual impairment; the absence of screen may also trigger this technique
Basic Flow	The text content can be ‘simply’ rendered with a speech tool, the video content can be presented with its audio part, and an image or animation can be described in terms of alternative text or description in an auditory format

Alternate Flows	

Use Case Name	**Direct Guidance (c.r. Suggest, Recommend)**
Use Case Description	This technique decides and presents to the user the next best option for interaction, according to the user's goal and other parameters being considered.
Primary Actor	User / System
Precondition	There is a navigation scheme, and a given criteria to present links in a certain order for the users
Trigger	The context of use, application domain, error rate, the level of expertise of the user
Basic Flow	Given a pre-defined criteria, the system suggests to the user the next steps that she should follow (criteria can take into account previous interactions recorded in log file, for instance)
Alternate Flows	

Use Case Name	**Change of Attitude**
Use Case Description	It means that parameters as the main intention of the communication, the intended affect or the level of emphasis could vary in order to modulate the message.
Primary Actor	User / System
Precondition	There is an avatar where the non-verbal and the verbal communication are expressed.
Trigger	The context of use, application domain, error rate, the level of expertise of the user
Basic Flow	When a problematic situation is detected then the system could decide to change the attitude of the avatar in order to avoid errors spirals, user's frustration, etc.
Alternate Flows	

Use Case Name	**Frame-based**
Use Case Description	Provide alternative content for the user that can be helpful to support the original goal (e.g. examples, links, several explanations)
Primary Actor	User / System
Precondition	The user task can be supported by additional content; there are alternatives available to be presented.
Trigger	The user has a complex task to be performed; the user is novice; the user is having troubles in accomplishing one task
Basic Flow	The alternative content is prepared to support the user task and it is provided a way to access it
Alternate Flows	

Use Case Name	Annotation of Links
Use Case Description	The links are augmented with the addition of notes, that remark their relevancy or context
Primary Actor	User / System
Precondition	There is a set of links that the end user can access while interacting
Trigger	The expertise level of the user, the error rate
Basic Flow	The links must be analysed and additional content created and related to them, the content is then presented to the end user
Alternate Flows	

5.1.2 Integration of the Serenoa framework

We propose the following example of adaptation (see Figure 6) in order to show the integration and role of the different Serenoa's components into the E-Health scenario. Thus the adaptation process performed by the Serenoa framework would be as follows: Jane starts using the desktop application, at home. When she is outside and she tries to access to 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. Afterwards the Context Manager warns about a high level of noise, the avatar was speaking hers answers but now it has no sense. So, again Runtime module is responsible to automatically stop the avatar voice feature. Finally, and due to the hard conditions for the interaction inside the bus, a more empathic attitude of the avatar is advisable in order to control the frustration of the user. In this case, the Generator module changes the avatar's mood trying to make friendlier the interaction.

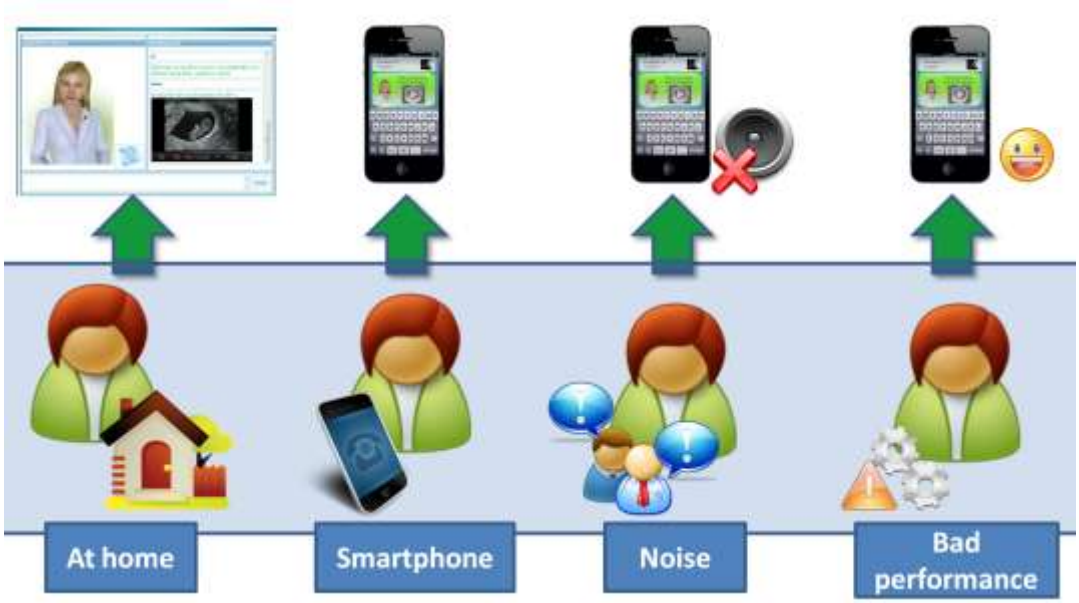

Figure 6: Adaptation example for E-Health scenario

5.1.3 Technical Realization

As it was already mentioned, the E-Health scenario is based on two existing e-health projects from TID: SARA project (chronic patients) and HealthDrive (patients' personal information). The technologies which were employed in the implementation of both interfaces had to be changed or adapted to let include Haptek avatar engine and to be enough flexible for a smooth integration into Serenoa framework.

Thus, our first implementation effort has been the development of the interface mock-ups corresponding to each of the scenarios considered. In Figure 7 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 embed and although it has text-to-speech capabilities, a text version of what it is said is also written in a text area (i.e. the white one in the middle of the interface). The buttons located at the bottom and the top of the interface give access to the rest of functionalities of the application (i.e. the agenda, the tests, the inbox, etc.)

Figure 7: Snapshot of the SARA project interface

Regarding the HealthDrive component, a snapshot of the current state of the conversational interface is given in the Figure 8. Three main areas have been conceived: the virtual assistant zone (left), the chat (right) and the text input (bottom). Notice that just next to the avatar there is an area which is reserved for the inclusion of buttons to let the user configure some aspects of the avatar. Currently, there is the possibility to turn off the speech output.

Figure 8: Snapshot of the HealthDrive project interface

Both developments abovementioned have been performed using HTML5, CSS and Javascript libraries as jQuery[2]. For the integration of the avatar, Haptek[3] provides an ActiveX component which is embed 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.

Regarding the adaptation of the avatar engine, the objective is to have a new runtime that we could use seamlessly with the Serenoa framework, but able to be rendered in a mobile phone, tablet, etc. Usually most of the avatars engines add high computational load and video render capabilities. Furthermore the technology (ActiveX, Flash...) that they use is often not compatible with most of the platforms. Thus, two alternatives have been explored:

1. A video-based one, using HTML5 video. Video chunks for predefined dialogue fragments could be produced, and put together in one single video file or else in multitude of video files accessible to the client application. The playback of these fragments may be done programmatically using the capabilities[4] in the <video> tag of HTML5, i.e., using the .currentTime property that allows us to fast forwards or backwards to one point in the video, hence providing access to particular snippets of action very quickly. More complex techniques are possible with the manipulation of the HTML5 Canvas defined by the <video> tag. For example, chroma key for transparent backgrounds[5], tracking of colours and shapes and appearance of other HTML objects inside the video frame[6] and others. The usage of third party media libraries, such as popcorn.js[7] or jPlayer[8] for jQuery allows easy integration of advanced functionality in the video controls, i.e., launching custom events for synchronization of activities in the web applications[9].
2. A simple, barely animated, 'cartoon-like' avatar engine, consisting on various avatar screenshots that

[2] http://jquery.com/
[3] http://www.haptek.com/
[4] https://developer.mozilla.org/en/DOM/HTMLMediaElement
[5] http://www.tinkernut.com/demos/273_chroma/video_chroma.html
[6] http://anavallasuiza.com/popcorn/
[7] http://popcornjs.org/
[8] http://jplayer.org/latest/developer-guide/
[9] http://yfrog.com/5onzwz

can be changed programmatically to reflect different attitudes, emotions or other communicative intents, and a 'comic balloon' for writing the text sentences that the avatar is supposed to communicate (see Figure 9). Again, this may be done using the capabilities of HTML5 and CSS3, which provide extensive new animation capabilities. Toolkits[10] are available providing animation primitives. Audio output for Text-to-Speech is still not in an advanced stage, but there are some ongoing efforts[11][12] that can be counted on for the future.

Figure 9: Mock-up for the HealthDrive project interface

5.2 E-Commerce transaction 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 on the **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.

5.2.1 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.

Use Case Name	**English speaking shopper experience / iPhone**
Use Case Description	A typical scenario for the front-end: an English speaking online shopper uses his iPhone to access the front-end application. He browses through the bike shop, selects 2 items and checks out. This case will show adaptation both to (1) device and (2) language
Primary Actor	Online shopper, English speaking, using an iOS based device (iPhone)
Precondition	Connection to the web site. iOS player installed on the iPhone
Trigger	The web server receives a session initialization request with indication of the device type (mobile) and the language to use (English)

[10] http://labs.hyperandroid.com/static/caat/
[11] http://syntensity.com/static/espeak.html
[12] http://code.google.com/chrome/extensions/tts.html

Basic Flow	Given the device type and the language, the e-commerce content adapts accordingly at display time
Alternate Flows	Same features available in different languages, on Android and from any web browser

Use Case Name	**French speaking color-blind desktop shopper experience / home computer (JQuery)**
Use Case Description	Another typical scenario for online shoppers, but with adaptation for color-blindness: a French speaking online shopper uses his web browser from his home computer to access the front-end application. He browses through the bike shop, selects 1 item and checks out. This case will show adaptation to (1) device, (2) language and (3) color-blindness
Primary Actor	Online shopper, French speaking, color blind, using a home computer
Precondition	Connection to the web site. The user indicates he needs adaptation for color-blindess
Trigger	The web server receives a session initialization request with indication of the device type (desktop) and the language to use (French). It also receives an explicit request to adapt colors.
Basic Flow	Given the device type, the language and the disability the e-commerce content adapts accordingly at display time
Alternate Flows	Same features available in different languages, on Android or iOS

Use Case Name	**English speaking customer representative experience / office computer (JQuery)**
Use Case Description	A typical scenario for the back-end customer representative: an English speaking employee (customer representative) uses his office computer to carry out his daily tasks on the back-end application. He browses through the employee portal, processes a pending order request issued by an online shopper and disconnects. During his session, he adjusts his user preferences This case will show adaptation to (1) device and (2) language and (3) user preferences
Primary Actor	An English speaking customer representative, connecting from his office computer web browser
Precondition	Connection to the web site Successful authentication requested as an employee
Trigger	The web server receives a session initialization request with indication of the credentials, device type (desktop) and the language to use (English).
Basic Flow	Given the device type, the language and the user preferences the content of the Intranet application adapts accordingly at display time
Alternate Flows	Same features available in different languages, on Android or iOS

Use Case Name	French speaking manager experience / office computer (Java-Swing App)
Use Case Description	A typical scenario for the back-end manager: a French speaking employee (manager) uses his office computer to supervise the current activity on the back-end application. He uses his computer office and starts a Java/Swing based application to visualize Business Intelligence data, displayed through charts (pie and bar charts) and dashboards. During his session, he adjusts his user preferences. This case will show adaptation to (1) platform, (2) language and (3) user preferences
Primary Actor	French speaking company manager, using a Java/Swing based App
Precondition	The user starts his rich client application (possibly through Java Web Start) and connects to the data server Successful authentication requested as an employee
Trigger	Authentication is successful and an environment variable allows to identify the working language (French)
Basic Flow	Given the platform type (rich client), the language and the user preferences the content of the Intranet application adapts accordingly at display time
Alternate Flows	Same features available in different languages, from a computer web browser, an Android or iOS mobile

Use Case Name	English speaking manager experience / Android
Use Case Description	A scenario for the back-end manager working remotely: an English speaking employee (manager) uses his Android mobile to access remotely company data for monitoring purposes to visualize Business Intelligence data displayed through charts (pie and bar charts) and dashboards. During his session, he adjusts his preferences on his mobile. This case will show adaptation to (1) Device, (2) language and (3) user preferences
Primary Actor	English speaking manager using an Android tablet
Precondition	Connection to the web site Successful authentication requested as an employee
Trigger	The web server receives a session initialization request with indication of the credentials, the device type (Android) and the language to use (English).
Basic Flow	Given the platform type (rich client), the language and the user preferences the content of the Intranet application adapts accordingly at display time
Alternate Flows	Same features available in different languages, from a computer web browser (or Java Swing App), or an iOS mobile

5.2.2 Integration of the Serenoa framework

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:

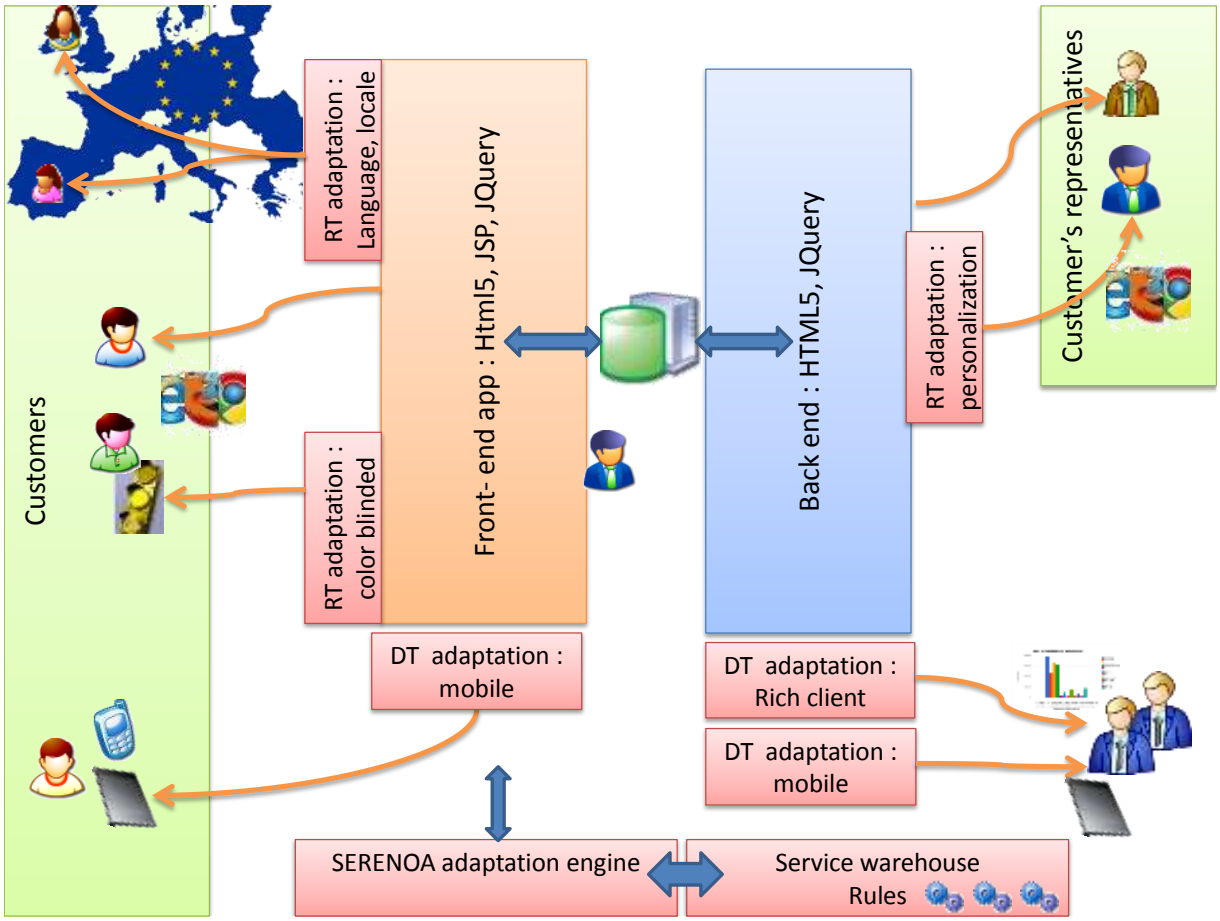

Figure 10: E-Commerce Prototype Architecture

In the above figure, 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

5.2.3 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,

Some people may also use smartphones or tablets: with the mobile device's internal web browser, it is sometimes not very user-friendly. Offering an application dedicated to mobile adaptation for the public could be an opportunity to gain attractiveness and to convey a modern image of the company.

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 BI features must be offered.

The following figure depicts how the E-Commerce scenario is technically realized:

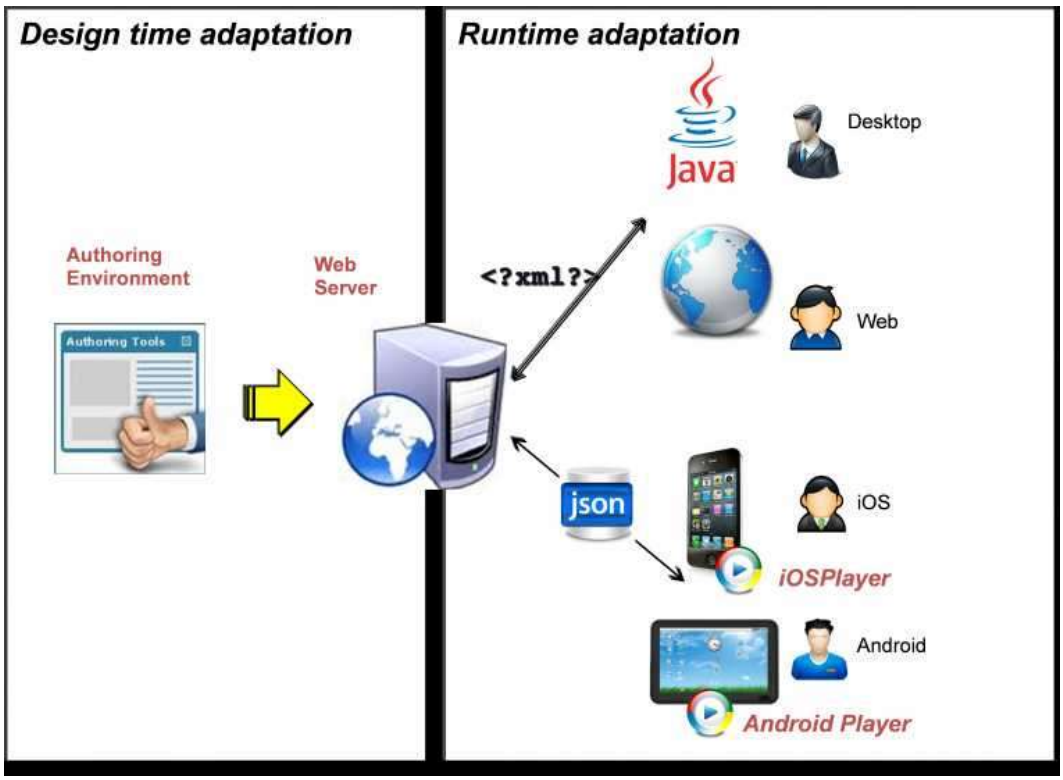

Figure 11: Technical Realization of E-Commerce Scenario

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.

5.3 Intelligent Picking Prototype

The objective of the warehouse picking system (WPS) prototype is to demonstrate more about its features of adaptive user interface. So instead of focus on business logics of system, what we more concern is the implementation of these potential adaptive rules for UI mentioned above. 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. However the coming 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. By giving up too much input actions, the self adaptive features enable user to finish the picking task more efficient and effective.

5.3.1 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 current version of the prototype and possibly not all will be implemented in the final version. Although all adaptations listed here are assumed to enhance the user experience of the interaction with the system, we need to evaluate this assumption in a user study to generate a final list containing the most reasonable and most desired adaptation use cases.

Use Case Name	**Minimize distraction (Fragile Item)**
Use Case Description	When the item to be picked is fragile, switch from multimodal to only-vocal modality.
Primary Actor	User / System
Precondition	The user has to pick a an item
Trigger	The item to be picked is fragile
Basic Flow	When the item to be picked is fragile, switch from multimodal to only-vocal modality, in order not to distract the user while picking the item.
Alternate Flows	

Use Case Name	Remind next action (Destination Shelf reached)
Use Case Description	Support the user when he does not know which item to pick
Primary Actor	User / System
Precondition	The user has reached the destination shelf
Trigger	The user does not confirm that he has picked the item for more than 45 seconds after reaching the correct shelf
Basic Flow	The application assumes that the worker is confused and s/he is not able to recognize the correct item to pick. Therefore, the application visualizes an image representing the item to pick, repeating the item identifier vocally, in order to help the worker in the item identification.
Alternate Flows	

Use Case Name	Switch off vocal modality (Noisy Environment)
Use Case Description	Switch of the voice interface when the environment is too noisy
Primary Actor	User / System
Precondition	The application is using both the graphical and vocal modality for interacting with the user
Trigger	The level of noise in the environment triggers this technique.
Basic Flow	The application switches to the only-graphical modality, since it is not possible to properly use the aural channel.
Alternate Flows	

Use Case Name	Optimize Path (Traffic Jam)
Use Case Description	The system changes the sequence of picking items to optimize the path
Primary Actor	User / System
Precondition	The last order is completed
Trigger	There are other orders in the list that contain items to be picked in the same room as the item just picked. The worker selected the path optimization preference (s/he wants to minimize the space covered for completing the different orders)
Basic Flow	The application re-arranges the order list, putting one of the orders that have items in the same room as the next element.
Alternate Flows	

Use Case Name	Suppress Information (Experienced Workers)
Use Case Description	Experienced users are shown more orders on one screen by omitting the navigational interface
Primary Actor	User / System

Precondition	The user is a warehouse expert (good knowledge of the warehouse organization)
Trigger	The user switches on the system
Basic Flow	The application hides the information about how to reach the different shelves, in order to show more orders in the same screen.
Alternate Flows	

Use Case Name	**Change Speed (Audio)**
Use Case Description	The audio content can be played in a higher or lower speed, according to the user's preferences as stored in the profile
Primary Actor	User / System
Precondition	There is audio content, an aimed speed (different of the original one) and it is possible to change it
Trigger	The user preferences as stored in profile (for instance if the user is accessing a content in a language other than her mother tongue, she may prefer to access it in a lower speed)
Basic Flow	Given an audio content, the player has its speed parameter set to a different value
Alternate Flows	

Use Case Name	**Translate Language (Audio & Text)**
Use Case Description	Given an audio content in a given language, it is translated to another one
Primary Actor	User / System
Precondition	There is audio, and there is a version of the audio content available in another language, or a method implemented that allows the translation
Trigger	The user preferences or the user profile
Basic Flow	There are versions of the content available in different language
Alternate Flows	

Use Case Name	**Change Brightness (all Content)**
Use Case Description	The brightness or luminance of the content is modified according to the context
Primary Actor	User / System
Precondition	There are images composing the UI, there is a function to adjust the brightness level
Trigger	The level of light of the environment, the user preferences
Basic Flow	Given a pre-condition, such as, if the light level in the environment is high, then the level of brightness of the UI is set to another value
Alternate Flows	

Use Case Name	Change Font Family (Text)
Use Case Description	The text content has its font family modified
Primary Actor	User / System
Precondition	There is a text content, and it is possible to modify its font family (by applying a transformation, or loading an alternative version of the content)
Trigger	The user preferences as stored in profile
Basic Flow	The content is selected, the font family is modified, the new version of the content is presented to the end user
Alternate Flows	

5.3.2 Integration in the Serenoa framework

The content map in Figure 12 shows the relationship between adaptive UI prototype and other supporting modules in Serenoa framework. Through the use of model-based descriptions of interactive applications the adaptive features of the UI can be realized. At design time the various initial versions of the applications are developed in terms of three concrete descriptions (vocal, graphical, and multimodal) in the ASFE-DL of Serenoa which is based on the MARIA language. In addition, the relevant adaptation rules are specified in terms of events, condition, and actions according to a language for adaptation rules which is described by AAL-DL. Adaptation rules are triggered by contextual events from a context server, which can depend on various aspects (user preferences, environmental changes, application-related events …).

At run-time we have an adaptation server that is able to communicate with the context manager server in order to gain information on subscribed events, as well as the interactive device in order to update the application according to the adaptation rules. The context server is also able to communicate with the applications, which can manage directly some contextual events according to their specifications.

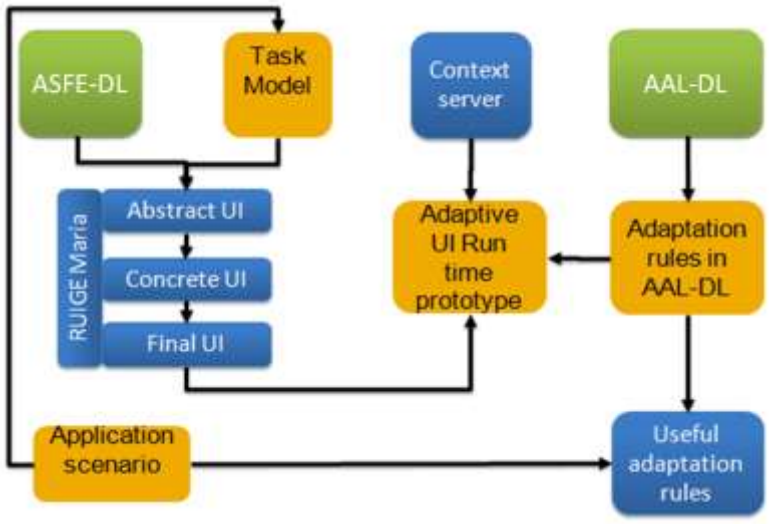

Figure 12: Content map of the adaptive UI application in Serenoa

5.3.3 Technical Realization

The UI component of the prototype is supported technically by HTML5, JavaScript and AJAX. AJAX enables the UI to keep a consistent layout and change partially. The navigation route in the Map view is drawn using the canvas label of HTML5. Speech recognition is realized using the speech input label of HTML5 and calling a respective API. The technical framework of the application implementation is shown

in Figure 13.

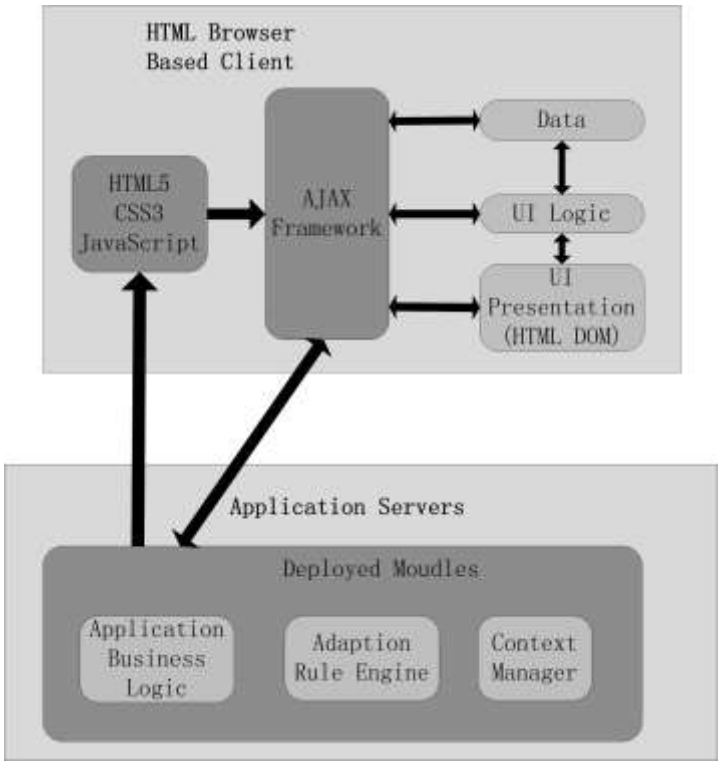

Figure 13: Technical Framework of Adaptive UI

Since there is no alternative interactive option except for audio input in user interface, the keyboard, mouse and touchable screen are not necessary. Based on the requirement and features of user interface, Head Mounted Device (HMD) and a wearable computer build the whole application. The architecture of the system is described in Figure 14. The adaptation server sends the updated data to the wearable computer after a change in the context has triggered the execution of an adaptation rule. The display is used for the visual output, the earphone for the vocal output and the microphone for the vocal input of the user. Some changes might be triggered by the smart environment (e.g. tracking of the picker's position or the item's location).

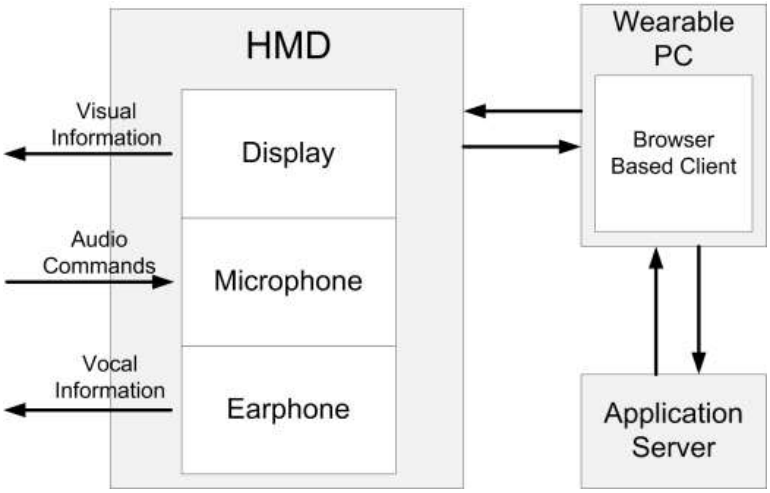

Figure 14: Adaptive prototype architecture

6 Conclusions

This document reports the current status of the development of the Serenoa application prototypes with a focus on requirements, design, technical realization and adaptation use cases.

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

- the scenarios
- the requirements and evaluation criteria
- an agile methodology description
- the status of the current prototypes
- the integration of the Serenoa framework in the prototypes
- adaptation use cases

In this deliverable, we had put much effort in aligning the style of reporting the requirements, evaluation criteria and adaptation use cases between all the partners with their own scenarios each. Compared to previous deliverables concerned with the requirements and evaluation criteria, we now applied a common style of representation (i.e. a table format) and assimilated the written explanations. Adaptation use cases are represented in a common table style as well.

The proposed agile methodology represents an identification of best practices and successful patterns already present in methodologies such as Scrum and UCD and their application to Serenoa's goals.

Summarizing, this document outlines the current state of the prototype development including requirements implementation and planned evaluation, shows how the Serenoa framework is (planned to be) integrated in their technical realization and presents scenario-specific adaptation use cases.

7 Future Work

Within the Task 5.2 Application Prototypes, the next deliverable is due in month 36. Until then, the prototypes (the systems respectively) need to be in their final versions. Future work will therefore concentrate on the further development of the prototypes. A constant effort will be made to integrate modules and concepts developed in the course of the Serenoa framework. 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 task T5.2 ‘Prototype Development’ will also contribute to and benefit from the works in task T5.3 ‘Evaluation’. Both tasks are closely related since the objects of interest are in both cases the prototypes and their requirements. The prototypes will be the object of evaluation, while the results of the evaluation will be used to further develop the prototype.

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 (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://files.morfeo-project.org/serenoa/public/deliverables/M12/-Serenoa_D2.4.1.pdf
10. 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://files.morfeo-project.org/serenoa/public/deliverables/M12/-Serenoa_D3.4.1.pdf
11. 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://files.morfeo-project.org/serenoa/public/deliverables/M12/-Serenoa_D3.1.1.pdf

Acknowledgements

- TELEFÓNICA INVESTIGACIÓN Y DESARROLLO, http://www.tid.es
- UNIVERSITE CATHOLIQUE DE LOUVAIN, http://www.uclouvain.be
- ISTI, http://giove.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/