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

Project no. FP7 – ICT – 258030

Deliverable D5.2.1 Application Prototypes (Req. and Design)

Due date of deliverable: 30/09/2011

Actual submission to EC date: 30/09/2011

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

Document Information	
Lead Contractor	TID
Editor	SAP
Revision	V1.0.1 (30/09/2011)
Reviewer 1	Telefónica I+D
Reviewer 2	
Approved by	Telefónica I+D
Project Officer	Jorge Gasós

Contributors	
Partner	**Contributors**
SAP	**Anna Lewandowski, Safdar Ali, Joerg Rett**
W4	**Jean-Loup Comeliau**
W4	**Jean-Loup Comeliau**
CTIC	**Ignacio Marín**
TID	**Javier Caminero Gil, Mari Carmen Rodríguez Gancedo**

Changes			
Version	**Date**	**Author**	**Comments**
0.1	06/06/2011	SAP	Initial draft with structure and contributors
0.2	24/06/2011	SAP, W4, TID, ISTI, CTIC	First contributions provided by partners
0.3	29/06/2011	SAP	Consolidation sent to partners and reviewers
0.4	04/07/2011	TID	Comments sent to partners
0.3	25/08/2011	SAP	Re-consolidation after significant changes sent to reviewers

0.4	29/08/2011	TID	Comments sent to partners
0.5	02/09/2011	SAP, W4, TID	Final contributions sent to SAP
0.6	09/09/2011	SAP	Final consolidation sent to reviewers
0.7	16/09/2011	TID	Comments sent to SAP
0.8	23/09/2011	SAP	Final version sent to TID
1.0.1	30/09/2011	TID	Final version send to EU

Executive Summary

This deliverable presents the first release of the SERENOA application prototypes with a focus on requirements and design. The document describes the three scenarios, proposed by the three industrial partners in which the prototypes will be deployed. The E-Health scenario of TID, the E-Commerce Transaction scenario of W4 and the Warehouse Management scenario of SAP are three completely different and independent scenarios. The scenarios can be seen as the „roots" of the application prototype „tree". Following this picture, the requirements derived from the general architecture and interviews with end-users and developers can be seen as the „trunk" which is aimed to be generic and general for all three prototypes. Additionally, the evaluation criteria are presented which have a close relation to the requirements. Finally, the three independent „branches" of the application prototypes are shown, through first design mock-ups and statements concerning the prototype development process.

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). Generic integration stands mainly for the discussions on common interfaces on the level of application development.
- Task T5.2 „Prototype Development" will investigate three prototype applications based on technologies developed in the previous work packages. Such instantiations in different domains and real-life scenarios will provide a soundness test of our technological developments and also evaluate their feasibility from a practical point of view. A first step toward prototype applications is a detailed design how to implement the example scenarios as introduced in WP 1.
- Task T5.3 „Evaluation" will evaluate the functionality, usability, effectiveness and quality of the developed technologies a practical perspective. In addition to the user-centred evaluation of the prototypes, this task also includes an evaluation of the introduced methodologies for context-aware SFE development.

In this first phase of task T5.2 „Prototype Development" the focus was on requirements and design for the application prototypes. The current state of the three scenarios was determined in order to continue the preparatory work of task T1.1 „Identification, elicitation and analysis of requirements". The previously gathered requirements were revisited in the light of the recently defined architectural specifications and criteria. The design mock-ups as well as the definition of development processes present the current state of the prototype development.

1.2 Related documents

- Deliverable D1.1.1 Requirements Analysis (R1) describes the scenarios and discusses the gathered requirements by means of the scenarios.
- Deliverable D1.2.1 Architectural Specifications (R1) provides useful indications about the project results that will have to be considered for integration.
- Deliverable D2.4.1 Criteria for the evaluation of CAA of SFEs defines a set of criteria and potential benchmarks for applications.
- Deliverable D3.4.1 Agile Methodology Description (R1) presents the first release of the SERENOA agile methodology description.
- Deliverable D4.5.1 Authoring Environment (R1) provides comparable results from the designers" viewpoints.

1.3 Organization of this document

Chapter 2 describes the three scenarios, proposed by the three industrial partners in which the prototypes will be deployed. The E-Health scenario of TID, the E-Commerce Transaction scenario of W4 and the Warehouse Management scenario of SAP are three completely different and independent scenarios. Chapter 3 presents the requirements derived from the general architecture and interviews with end-users and developers which is aimed to be generic and general for all three prototypes. Additionally, evaluation criteria are presented which have a close relation to the requirements. In chapter 4, the three application prototypes are shown, through first design mock-ups and statements concerning the prototype development process. The deliverable ends with a conclusion, a summary and an outlook on future work in chapter 5.

2 Scenarios

In this chapter we will describe the three scenarios of the SERENOA project. Whereas in D1.1.1 "Requirements Analysis (R1)" first ideas of UI developer and consumer scenarios were presented, here we will examine them in more detail and try to provide more concrete examples of usage. The first UI mock-ups emerging from these scenarios and our expectations concerning the user experience will be presented later in chapter 4. One scenario did not undergo any changes, while one was slightly modified and extended and the third has been completely rewritten.

Scenarios are living documents, which can be modified at any point of time and get more detailed as the project unfolds. They 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.

At the time of writing of this document, some of the SERENOA scenarios may differ from each other in their level of detail, which is not uncommon. According to Rosson and Carroll (2002) scenarios are both flexible and concrete. Depending on the project status the scenarios get more and more advanced or remain in their level of detail.

In SERENOA each industrial partner has designed their own scenario in which the prototypes will be developed:

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

Each partner develops a prototype for his particular story with its specific end-users and consumers of the application. TID‟s consumers are firstly a normal person living in a digitally augmented smart home and secondly a patient recently released from medical care or suffering from a chronic disease that requires frequent medical supervision. Consumers for W4‟s prototype are customers of an E-Commerce website, who can create an account and enter their payment details, authenticate themselves by using their username and password and make a payment. SAP‟s consumers are warehouse clerks who work in different environments, use different devices – for instance desktop computer, mobile and head mounted display (HMD) – and need to orientate themselves in a big warehouse while picking parts from big shelves.

In addition, the possible usage of MyMobileWeb[1] will be studied to show how different runtimes using the Serenoa architecture are interoperable. Interoperability means, in this context, that using the same data model and parts of the UI interface definition and other application definition aspects we can create new applications (a new "flavor" of the application for another runtime or a port of such already existing application from one runtime to another). The exact prototypes created making use of MyMobileWeb have not been defined yet, as they depend on a deeper definition of both the SERENOA architecture and the original prototypes from the three partners mentioned above.

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 program provides 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

[1] http://mymobileweb.morfeo-project.org/ MyMobileWeb framework.

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[2] 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 ongoing 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.

The SERENOA Telefónica I+D group is going to provide these two pilot projects with a new concept 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 Advanced E-Commerce Transaction Scenario

The goal of W4‟s SERENOA prototype is the implementation of an E-Commerce scenario that illustrates various auto-adaptive features 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 to the practice of making web sites usable by people of all abilities and disabilities.

Here, we will not deal with disabilities such as blindness or deafness, which require more sophisticated hardware devices 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 handle 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. After that to make a payment, they authenticate by simply using their username and password. The company earns revenues from keeping a fraction of the transaction amount and serves as an intermediary between the product providers and the customers. Therefore, it has no product stock. 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. A more detailed description of the E-Commerce scenario can be found in deliverable D1.1.1.

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:

- Both Front-office and Back-office systems will not consist of “real” operational E-Commerce systems, but rather of showcases (products are available to implement such architectures as well as sophisticated shopping cart policies);
- A simplified item catalog will be made available for the purpose of the demo;
- Both Front-office and Back-office users will be registered in some data system with their specificities.
- The intent here is not to offer a comprehensive workflow by exploring all the branches of the process

[2] Consejería de Salud de Andalucía: http://www.ucm.es/cont/dbinformaciondescargas/101/4_q5.pdf (Spanish only)

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.

2.3 Warehouse Management Scenario

The scenario for SAP‟s SERENOA prototype is aimed at providing a seamless context (environment and task) adaptation experience to users in one of SAP‟s Living Lab facilities in the field of future retail concepts.

The Future Retail Centre (FRC) of SAP Research in Regensdorf, Switzerland shows over two floors demonstrators along the supply chain. The 21 demos are grouped into three scenarios Logistics, Retail (Sales) and Retail Management.

This organization reflects the logistic processes of the real world, where goods with integrated RFID tag labelling will be shipped from the place of production to a distribution centre. In the distribution centre goods are received, picked and issued to the store. As already described in deliverable D1.1.1 "Requirements Analysis (R1)", SAP‟s warehouse scenario is located in this environment. In this scenario, 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. The picking process makes high quality and time-constrained demands on the workers. The Logistics scenario shown at the FRC has its demonstrators arranged in form of a supply chain (see Figure 1):

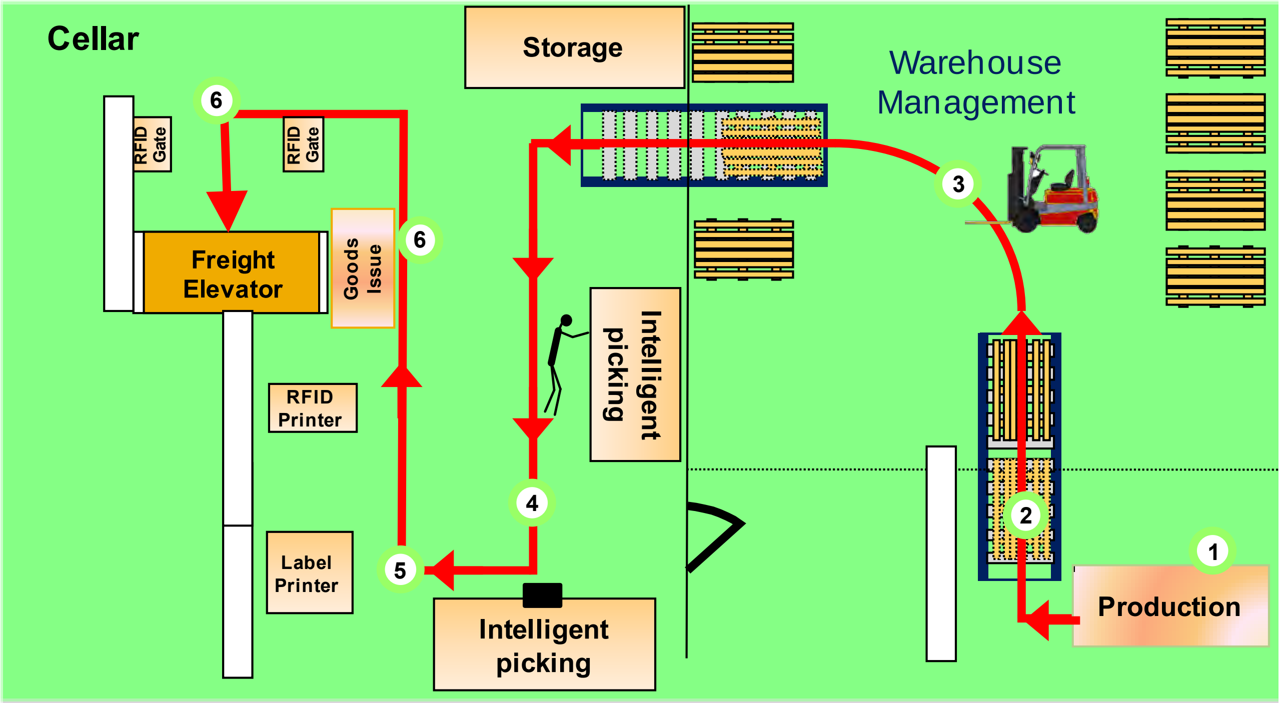


Figure 1: The Logistics scenario shown at the FRC: (1) RFID based Production with automatic material and production flow steering, (2) RFID Temperature Tracking with Supply Chain temperature Sensor, (3) Warehouse Management with an RFID-Forklift, (4) Intelligent Picking, (5) RFID Labelling trade units, (6) Pack and Ship & Good Receipt with RFID goods issue and palette aggregation with RFID hand reader.

The SERENOA prototype will enhance the demonstrator for intelligent picking (4). This scenario motivates how proactive applications can provide unobtrusive and adequate help (e.g. missing parts, location of necessary parts, etc.) when the user needs help. Thereby, the service time can be reduced while increasing the quality of service.

In the real world, the logistic process continues to the store, where the goods are received and smart store processes are applied. At the FRC, the supply chain continues to Retail Management with five demos, e.g. on price strategy and smart vending and concludes in Retail with eight demos, e.g. on Mobile Payment and RFID Shelves. This opens up the possibility while the SERENOA prototypes are evolving, to create further prototypes moving from Logistics to Sales scenarios.

3 Requirements, architecture and evaluation criteria

This chapter presents the requirements, architecture and evaluation criteria

In deliverable D1.1.1 “Requirements Analysis (R1)” requirements have been gathered from different sources, namely: analysis of interviews and observations done with the end-users, the results of market reviews, generic use cases elicited both by end-users and by the consortium. In general, requirements are identified to meet the end-user as well as the business needs.

In deliverable D1.2.1 “Architectural Specifications (R1)” the first version of the SERENOA architectural specifications for a framework enabling context-aware adaptive Service Front Ends was presented.

In deliverable D4.2.1 “Criteria for the evaluation of CAA of SFEs” the evaluation criteria were presented which will allow the access to the proposed solutions both from the designers and the end-user viewpoints in terms of their effectiveness and satisfaction. The criteria consider usability of various project results, including the adaptation at run-time.

At the end of the project, there will be a comparison between the initial list of requirements and project achievements through evaluation criteria. At that point we shall be able to tell if the requirements were fully, partially or not implemented.

Having the evaluation criteria which themselves evolved through the deliverable D1.1.1 and deliverable D1.2.1 the necessary features of an application prototype can be defined. In an agile setting as described in deliverable D3.4.1 “Agile Methodology Description” the person in the role of a Product Owner (see D3.4.1) would set these definitions. As requirements, specifications and criteria evolve continuously along the project such would the Product Owner adapt his list of features.

The set of requirements from architecture, requirements from end-users and evaluation criteria is now summarized as a first starting point for the list of features of the application prototypes.

3.1 Requirements from architecture

The architecture of SERENOA was defined in the project deliverable D1.2.1 and is in a descriptive stage in which the main lines for future implementation of modules and their relationships are laid out. However, in order to inform future developments and implementation details we may propose here a list of requirements from the architecture by the prototypes described. The following list shows requirements from such an architecture:

- **Developer feedback:** For documenting and debugging matters, user interaction needs to be recorded by using log files and user history features;
- **Verbosity Control:** The architecture should be able to provide feedback such as the described above in various levels of verbosity and depth. E.g., while testing connectivity between modules details upon the HTTP connections might prove useful, while they would be only clutter when testing adaptation at a higher abstraction level;
- **Context Awareness:** The architecture needs to accommodate a context infrastructure that allows the SERENOA system to be aware at all times of all the possible details of the active context in which the system is running. By context we refer to data upon the active user, the platform executing the SERENOA application and the environment (location and associated conditions) in which the user is executing the system. Further detailing the list of requirements, this context information needs to be **Accurate**, **Timely** and **Robust**. As such, the modules in the architecture needs to provide correct data, has to do it at the appropriate time without unnecessary delays and has to be resilient to changes in operation or errors from the sensing equipment;
- **Modularity:** The architecture should be kept modular. This will allow us to test different pieces of software fulfilling the same purpose in order to check their performance or other development metric;

- **Transparent:** The architecture in itself is not the target of the system, so its details should not be an integral part of the SERENOA concept, e.g., it should be conceptually possible to change from a client/server based architecture to a message passing one in the event that it proves to be more appropriate. Note that this is not equivalent to stating that the architecture will be able to provide this functionality out of the box, but that the architecture will serve SERENOA and will be constructed around its requirements, not the other way around;
- **Performance:** The architecture needs to be optimized to enable an efficient working of the system and not introduce significant overhead on top on SERENOA-essential computing;
- **Simplicity:** Related to the requirements above, the architecture should be conceptually simple enough so that the development process is not encumbered by the details of implementation of modules for the architecture. Development work should be devoted in its majority to solving SERENOA problems, not architectural issues.

3.2 Requirements from users

In SERENOA the overall system, end-user and business requirements (functional and non-functional) for multidimensional context-aware adaptation of SFEs were identified and elicited. The end-user approach is specifying the need for such a system to be used by end-users or consumers, while the business requirements aim towards the needs of businesses to offer such a platform to their customers and therefore create more revenue from their business. Finally, the software approach aims at the requirements from a software-centred approach, which includes non-functional requirements that must be satisfied by the target architecture.

Requirements in software and usability engineering are needed to describe how the system should be, how it should behave and what it should do.

3.2.1 Requirements from developer scenarios

Though for application prototypes the focus will be on requirements from end-users of type “consumer” some requirements from the type “developer” should also be regarded. In this sense requirements described in D1.1.1 as “The developers should build applications able to ...” will be transformed to “The application should be able to...”:

- **Multimodality**. The applications should be able to support other modalities apart from the graphical one.
- **Cross-platform consistency**. The applications should provide the same experience on different platforms for users.
- **Accessibility**. The applications should be able to support the need for users to have better accessibility.
- **Independence of Different Technological Spaces**. The developed system should be platform independent. The user wants the system to work on different platforms or devices and there should be consistency between them.
- **Platform-dependent Adaption**. The applications should be able to dynamically adapt to the different characteristics of platforms.
- **User-dependent Adaption**. The applications should be able to adapt according to the different characteristics and needs of users.
- **Environment-dependent Adaptation**. The applications should be able to adapt according to the different characteristics of the environment.
- **Should Support Several Adaptation Techniques**. The applications should be able to support not only automatic adaptation but also adaptation specified by the user (by means of e.g. options selected in menus, etc.).

3.2.2 Requirements from E-Health scenario

Some requirements related to the E-Health scenario have been presented in deliverable D1.1.1 through the discussion on smart Home Tele-assistance with the TID-team:

- **Audio Feedback,** particularly Text-to-Speech synthesis but also in non-verbal events, is key in human-avatar communication and will be a crucial aspect in adaptation.
- **Visual Feedback,** as in the availability and usefulness of non-verbal communication from the avatar and special effects such as focusing the viewport on a part of the avatar's body or the use of lightning to assist in the turn-taking process.
- **Multimodality**, as it refers to the ability to successfully combine several modalities of communication in a working system. In the health system, the use of medical data such as graphs or documents along with the actions of the avatar will be investigated and its adaptation worked on.
- **Customization.** Avatar-based systems pose particular customization challenges, such as different virtual embodiments in factors such as general physical aspect, synthesized speech qualities, gender and even avatar clothing and gesturing.
- **Control Over the Adaptation Process.** Users may prefer to dispose of the avatar completely and/or use just the speech version without a physical visualization of the character.
- **Cross-Platform Consistency,** as the avatar engine won't easily be available with the same characteristics in devices across the board, extra effort will be put in delivering a consistent avatar experiences 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.
- **Satisfaction,** as metrics involving the avatar will be taken into account as feedback both in the design process and in runtime, as avatars are interfaces which can interact with the user to get on-line feedback that can be used to affect some runtime parameters (e.g., number and verbosity of avatar appearances).
- **Search** in order to provide the health application with search capabilities.
- **User-dependent Adaption**, as the system, including the avatar, will need to be adapted to different configurations for the need of different users.
- **Anticipation of Events**, as the prototype will need to track situations so it can anticipate issues that may trigger adaptation.
- **Accessibility**, as to keep the prototype accessible for users with motor and/or cognitive disabilities including elderly people not used to technology.
- **System and Task Continuity**, as the prototype should be able to make the user aware of the different tasks to be performed in each user session. The avatar will be used to verbally instruct the user and also strategies of persuasion will be used in some tasks that may not be entirely pleasant for users (e.g., attending medical appointments, taking regularly medication).
- **Dynamic User Choice**, as should be able to select their preferred UI configurations easily for the system, crucially including aspects relating to the avatar presence and/or actions.
- **Intuitiveness,** as 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.
- **User trust,** as trust is a key aspect in any medical application and it is a key driver in our decision to use avatars. 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. Experience shows that unrealistically raised expectations and confusion upon whether there is a 'man behind the

curtain' can be a problem in avatar systems.

- **Confidentiality and integrity of data**, as the system is poised to use sensitive patient data such as medical reports and images. The data should be kept secure so breaches in the system don't compromise the patients' personal files.

3.2.3 Requirements from the advanced E-Commerce transaction scenario

Based on the two interviews conducted the W4-team, the most important requirements for context-aware adaptation of service front-ends can be sorted out in the following categories:

- **Easy Connection and Configuration**. The applications should enable the End users to configure easily their environment
- **Cross-Platform Consistency**. Adaptation to mobile devices, and more particularly smart phones
- **Platform-dependent Adaption**. Adaptation to mobile devices.
- **Multimodality**. Adaptation must translate into different interaction modes.
- **Independence of Different Technological Spaces**. Some features need to be automatically available on different platforms and specific features need to be activated based on the platform where the device is used
- **Accessibility**. Web accessibility.
- **Time to Market**. Shortening time to market remains a top priority.
- **Deadline-driven**. Development is often deadline driven

The feedback came from persons who are business driven managers, and who themselves manage teams that work on front-end applications.

3.2.4 Requirements from the Warehouse Management scenario

Based on the interviews conducted by the SAP-team, the most important requirements for context-aware adaptation of service front-ends can be sorted out in the following categories:

- **Stability**. The warehouse clerks need a stable system
- **Error**. The warehouse clerks need a system that shows them when a mistake was made and how they can correct it.
- **Response Time**. Run-time algorithms cannot stop the user from interacting in a fast way.
- **Performance**. Users want to perform fast without making mistakes
- **Efficiency**. Users want to perform efficiently without making mistakes
- **Satisfaction**. When the system increases the quality of work and satisfies the user, so that additional value to users‟work is provided, the worker is satisfied
- **Intuitiveness**. The workers are often part-time employees, therefore the system needs to be intuitive
- **Simplicity**. The workers are often no technical experts, therefore the system needs to be simple.
- **Learnability**. The workers shouldn‟t put much effort in learning how to use it
- **Identification**. The user needs an identification code with which he can access the system easily.
- **Easy Connection and Configuration**. It should be easy to connect to the system and to configure it
- **Working environment**. The system needs to be adaptive to the working environment
- **Independence of Different Technological Spaces.** The system needs to be adaptive independently of the different technological spaces

- **Cross-Platform Consistency.** The user experience on the different platforms should be similar and coherence should be guaranteed
- **System and Task Continuity.** The adaptation should run in a continuous process.
- **Control Over the Adaptation Process**. The user should have the control over the adaptation process.
- **User-dependent Adaption.** The system needs to adapt automatically to the user´s tasks, needs and capabilities.
- **Multimodality.** By saying “next” or performing the corresponding gesture the system should show the next screen to the user.
- **Visual Feedback**. The system should give immediate feedback to the user, if he picked the correct or wrong parts.
- **Accessibility**. The system should support different languages.
- **Customization**. The user should be able to customize the system by changing the settings.
- **Personalization.** The user should be able to personalize the system by changing the settings.
- **Memorability.** It should be easy for the user to perform tasks, when returning after a long period of time not using the system.

3.2.5 Requirements from Business goals

Some business goals for the SERENOA project have been presented in deliverable D1.1.1 which emerged from a deeper investigation in deliverable D6.1.1 “Exploitation Plan (Initial Version)”. Each partner will rate these business goals specifically for their company or institution.

- **Differentiating factor.** Make the User Interface become a strongly differentiating factor of a software package.
- **Develop strategic IP.** Perform a strategic positioning.
- **Reduced development costs.** Reduce the cost of development of a software release
- **Reduced maintenance cost.** Reduce the cost of maintenance of a software release
- **Reduced porting cost**. Reduce the cost of porting a software release to a different platform
- **Reduced time to market.** Reduce the time to market of software releases and upgrades
- **E-participation and e-inclusion.** Provide technology that will favour e-participation and e-inclusion
- **Innovation.** Make the EU investments be fruitful
- **According with norms and regulations.** Act in accordance with norms and regulations
- **Reduced process cost.** Improve the relative cost position of a certain process or activity
- **Best practices.** Implement the industry best practices
- **Expanded number of clients.** Expand the share in the Business Application markets
- **Expanded number of users.** Expand the user base of currently sold applications
- **Increased end-user experience.** Increase the richness of the application experience for the end-user (e.g. consumers of an E-Commerce website)
- **Increased end-user productivity.** Increase the productivity of the application for the end-user (e.g. business-users with a certain business task to accomplish)
- **Compatibility with market leaders.** Be compatible with the technologies, platforms and standards

with the highest market share or growth rate in the industry

3.3 Evaluation Criteria

As mentioned in deliverable D4.2.1 the evaluation criteria will allow the access to the proposed solutions both from the designers and the end-user viewpoints in terms of their effectiveness and satisfaction. The criteria consider usability of various project results, including the adaptation at run-time. The adaptation process will also be evaluated in terms of software engineering parameters (such as robustness, efficiency, portability, etc.).

The adopted solutions can be evaluated from two points of view: by using some relevant software quality factors (so doing a technical evaluation), and also by considering criteria aimed at evaluating user-oriented aspects (e.g. usability and accessibility evaluation).

The project results to be expected from the work in task T5.2 “Prototype Development” has been defined in deliverable D2.4.1 as “Applications”. In general, these results will be subject to both a technical evaluation and a usability evaluation. For the application prototypes mainly the user type “end-user” will be considered as in contrast to the e.g. authoring tool where the user type “developer” will be considered. Applications have been distinguished as:

- Adapted applications produced by authoring tools.
- Adapted applications produced by runtime adapters.

In both cases the attention will be on an end-user evaluation in terms of usability and accessibility.

3.3.1 Technical criteria

The following features relate to the technical evaluation:

- **Interoperability**, as the capability of the application to interact (inter-operate) with one or more other specified systems.
- **Correctness**, as the degree with which the application performs its tasks as defined in the requirements specification.
- **Adherence to standards**, as the ability of the application to comply with standards, regulations, guidelines, conventions, etc.
- **Security,** as the capability of the application to protect information and data so that unauthorized persons or systems cannot read or modify them and that only authorized persons or systems are allowed to access them.
- **Efficiency**, as the capability of the application to provide appropriate performance relative to the amount of resources used under stated conditions and to guarantee a specified level of performance when used under specified conditions.
- **Maintainability and changeability**, as the abilities of the application to be modified after a working version is delivered.
- **Extensibility and evolvability**, as the ability to extend/change a system so at minimum effort cost.
- **Modularity**, as the partitioning of the application design that allows complex applications to be manageable for the purpose of implementation and maintenance and that enable parallel work.
- **Reliability**, as the capability of the application to maintain a specified level of performance when used under specified conditions.
- **Availability**, as the characteristic how often an application is operational.
- **Fault tolerance and robustness**, as the capability of the application to maintain the availability of the service even in case of errors or failures.

- **Scalability**, as the ability of a system to handle growing amounts of work in a graceful manner or its ability to be enlarged to accommodate that growth.
- **Testability**, as the capability of the application to allow modified software to be validated.
- **Recoverability**, as the capability of the application to re-establish a specified level of performance and recover the affected data in the case of a failure.

3.3.2 User-oriented criteria

In the following a number of criteria aimed at evaluating more user-oriented aspects (e.g. usability, accessibility) are listed. It is worth noting that all such criteria (both technical and user-oriented) are very often dependent on each other. The Product Owner of an application needs to take into account such relationships and decide in case of contradiction which features receive a higher priority.

- **Effectiveness**, as the accuracy and completeness with which certain users can achieve specified goals in particular environments.
- **Efficiency**, as the resources expended in relation to the accuracy and completeness of goals achieved.
- **Satisfaction**, as the comfort and acceptability offered by the work system to its users and other people affected by its use.
- **Learnability**, as the capability of the application to enable the user to easily learn its application.
- **Memorability**, as the ability for users to go back to the system and remember how to use it once they have been away from it for some time, without having to perform relearning.
- **Comprehensibility**, as the capability of the application to enable the user to understand a) whether the application is suitable, and b) how it can be used for particular tasks and conditions of use.
- **Error tolerance**, as the ability of the application to provide users with relevant support in case of users‟errors, in order to allow an easy recovery.
- **Accessibility**, as the degree to which an application, device, service, or environment is available to as many people as possible, including persons with some level of impairment.
- **Attractiveness,** as the capability of the application to be attractive to the user.
- **Controllability and programmability** refer to the degree of control the application provides to the user.
- **Flexibility**, as the multiplicity of ways the user and the system exchange information (Input/Output) in different forms without fixed task ordering.
- **Feedback**. The system should offer informative and effective feedback about the effect of the interaction.

3.3.3 UI adaptation criteria

The previous criteria (both technical and user-oriented) can already be applied to the existing demonstrators which will be used to some extent (depending on the partner) during the preparation phase as the base for the application. In the following some criteria specifically relating to the aspect of UI adaptation will be presented:

- **Predictability of UI adaptation**, as the ability of the user to understand under which circumstances adaptation takes place, what UI parts are going to be adapted and where the user interaction results will appear after adaptation.
- **User's Awareness of UI adaptation**, as the ease of the user to realize the changes in the UI due to adaptation.

- **Appropriateness of the adaptation**, as the judgment of the user when he realizes that adaptation has been performed that the performed adaptation is appropriate.
- **Timeliness of the adaptation**, as the application of adaptation in a timely manner (e.g., not too late) when there is an actual need to change some aspect of the user interface to better support the user.
- **Controllability/Programmability of the adaptation**, as the user's control over the adaptive process, namely the users‟ ability to control both the circumstances that lead to triggering adaptation, and how adaptation is actually applied.
- **Perceived Usefulness of the adaptation**, as the user's perceived usefulness of the adapting system.
- **Within-device consistency of UI adaptation**, as the consistency of the UI design after adaptation with the design before adaptation.
- **Across-device consistency of UI adaptation**, as the level of consistency between the UI design after and before an adaptation following a device change. Device change implies that the user interface is migrated from one device to another.
- **Adaptation Performance**, as the time required to perform the adaptation.
- **Adaptation Transition**, as the behaviour of the user interface during the adaptation which should be understandable and should allow users to realize what is happening.

It is quite likely that the Product Owner of the application prototype will focus on user-oriented aspects. The relationship between the evaluation criteria and the requirements and scenarios is sometimes a direct mapping. For instance, the “Learnability” requirement (see deliverable D1.1.1) has a direct relationship with the “Learnability” criterion in the usability evaluation (see deliverable D2.4.1). The same holds also for other requirements (see for instance portability, efficiency, accessibility, etc.). In other cases requirements identified in D1.1.1 cover aspects that have not been identified in this document as specific evaluation criteria (see for instance the requirements connected with the Agile methodology, reported in D1.1.1). In other cases the opposite is true: see for instance the whole number of different criteria connected with UI adaptation that have been identified in this document, and which do not always have a direct counterpart in the project‟s requirement list.

The Product Owner of an application prototype should also consider how an evaluation of the application could be carried out. Usability evaluation can be carried out following various approaches. In the project, we plan to apply some of them. In inspection-based evaluation one expert analyses the user interface in order to detect potential issues. In user-based evaluation users are directly involved in detecting the usability problems; often this is performed through user tests.

4 Design of the Prototypes

Some of the prototypes will be developed upon already existing systems, like the HealthDrive SFE of TID to tackle the problem of context dependent adaptation. In the case of TID"s prototype for the HealthDrive system the adaptation of Embodied Conversational Agents (ECAs) also known as Avatars, to multiple devices, like smartphone or Tablets is the goal. SAPs prototype for intelligent picking also aims at multiple devices, additionally a smart sensor environment and user profiles are regarded. In the case of W4"s E-Commerce scenario, the system will initially be developed as a standard web application granting access to different users with different profiles and will be adapted progressively by defining and implementing adaptation rules for mobile devices. During the preparation phase reported in this deliverable the existing demonstrators have been evaluated towards their usefulness as the base for the application prototypes. Besides that, first mock-ups have been designed and the prototype development processes have been established. The following sections divide into the three application prototypes and on a lower level into the preparation of the existing demonstrators, design mock-ups and the development process.

4.1 Device dependent adaptation of Avatars by TID

Telefónica I+D is currently working on the definition of the prototype working in parallel with the designers of the backend and the users of the system (doctors and patients). As exposed in section 2.1, the work is separated in two distinct lines of work. We will focus here on the one (HealthDrive) which is more advanced; this scenario provides access to the Andalusian Health System user data by end users with SFEs augmented by the use of avatars and supporting multiple device adaptation.

In order to enable patient data to be available to online users, effective SFEs need to be developed to facilitate the user handling of this process. For having a preliminary assessment of this, the health system has begun a pilot project with Telefónica I+D to build a preliminary user front-end for a small population of test users. When this is operative, the prototype will be run for some time in order to gather feedback that will be useful when designing a possible full-scale implementation. The test system is to support desktop computers, tablet PCs and smartphones from its start.

Besides a traditional, WIMP paradigm based style of interactions, Telefónica I+D decided to invest also in using Embodied Conversational Agents (ECA) (Cassell J. S., 2000), commonly known as „avatars", in order to provide a more manageable interface for people not used to computers and to explore other interaction modalities. ECAs have been proven to provide boosts in user trust (Cassell & Bickmore, 2000) and also in the persuasive capabilities (Cavazza, 2010) of the system, both of which are significant benefits in a sensitive field such as E-Health. In addition, the latest research in ECAs (Danieli, 2010) has been able to successfully merge emotional handling and production. This is still a field where research is needed to assess the benefits and therefore has been deemed as suitable for a research pilot project such as this.

The integration of an avatar in a multi-device system, however, is a complex problem. In addition to the technical obstacles (e.g., engines that are suitable for different computing environments are scarce) there are clear interaction differences based upon the platform details such as for example:

a. TV-based applications, where the avatar is presented in close to life-like size

b. Desktop web applications, where the avatar is rendered inside a browser window in a smaller size

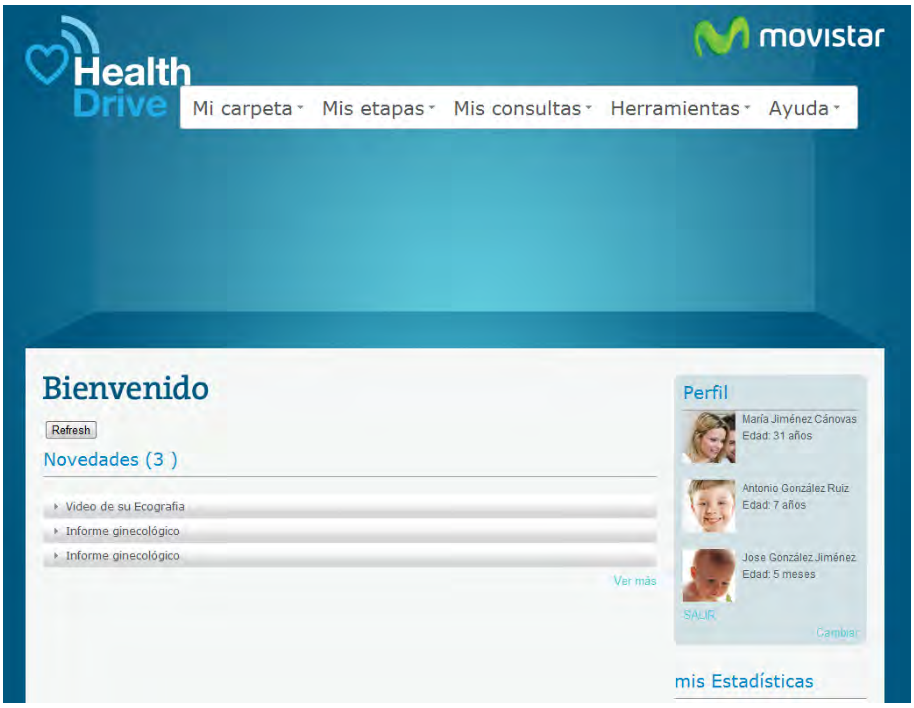
c. Phone applications, where the avatar is rendered in a very small size

For an ECA, size matters. ECAs are usually designed to be rendered at close to life-like (Cassell J. S., 2000) sizes where aspects such as lip-sync, facial gestures and body language can be interpreted in ways close to human-human communication. However, environments such as a smartphone have screen sizes around an order of magnitude smaller. This has a clear impact: precise lip-sync becomes less relevant and nuances in gestures begin to be difficult to discern. In addition, the screen size often dictates that the avatar is not able to be present at all times in the UI. This diminishes the impact of the ECA and reduces the sense of „presence" in the user.

Thus, in this SERENOA prototype, we are going to take advantage of the project achievements in adaptation to work on SFE adaptation for ECAs for multiple devices to provide solutions to these problems. In the remainder of this section we will summarily describe the actual state of the prototype and some ideas for the future development that will be used in SERENOA.

4.1.1 Current status of the SFE

The SFE for the system is web-based, and the current status is shown in Figure 2.



Bienvenido

Novedades (3)

- Video de su Ecografia
- Informe ginecológico
- Informe ginecológico

Perfil

María Jiménez Cánovas
Edad: 31 años

Antonio González Ruiz
Edad: 7 años

Jose González Jiménez
Edad: 5 meses

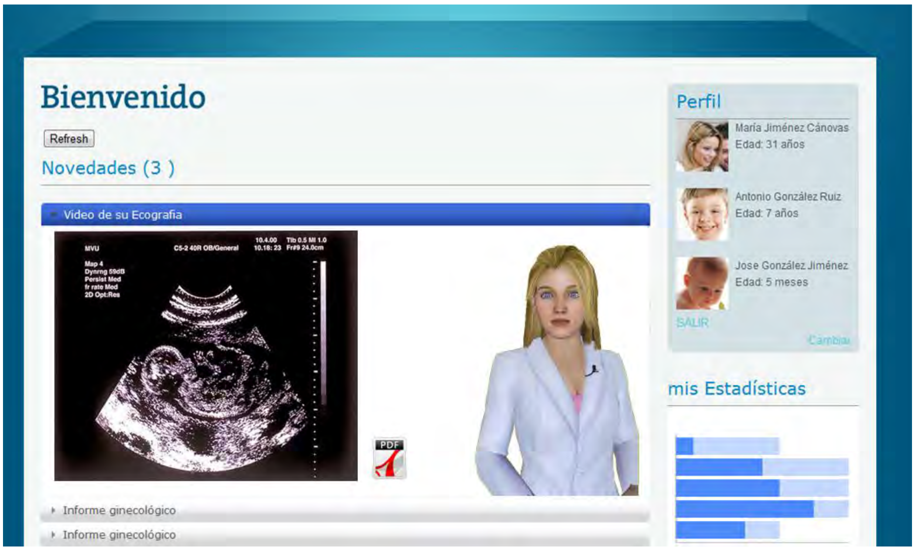
mis Estadísticas

Figure 2: HealthDrive Virtual Assistant screenshot: Start-Up screen (Spanish). Top of screen: Fixed menu bar with shortcuts to the various sections. Main section (under ‘Bienvenido’): Start-up screen in which the latest news as collapsible panels (e.g. ultrasound video). Right panel: User’s profile and hers/his ‘associated persons’ (e.g. user’s children)

Although this UI is just a development snapshot, the interaction and usability have been designed by Telefónica UX engineers and is expected to remain stable during the coming development phase. The SFE is currently a web application that uses JavaServerFaces[3] to dynamically build a UI for each user and each situation. Now we will present a series of screenshots of partially mock-up interfaces that appear upon interacting with the GUI

In Figure 2 the latest changes in the user‟s profile (labelled *Novedades*) are shown as a start-up screen. Upon clicking on the first headline (marked as *Ecografia-ultrasound scanner*) we can see more details as shown in Figure 3. Note we will now show just the bottom part of the interface, as the top part just shows the same menus anchored at all times:

[3] http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html : Oracle JavaServerFaces



María Jiménez Cánovas
Edad: 31 años

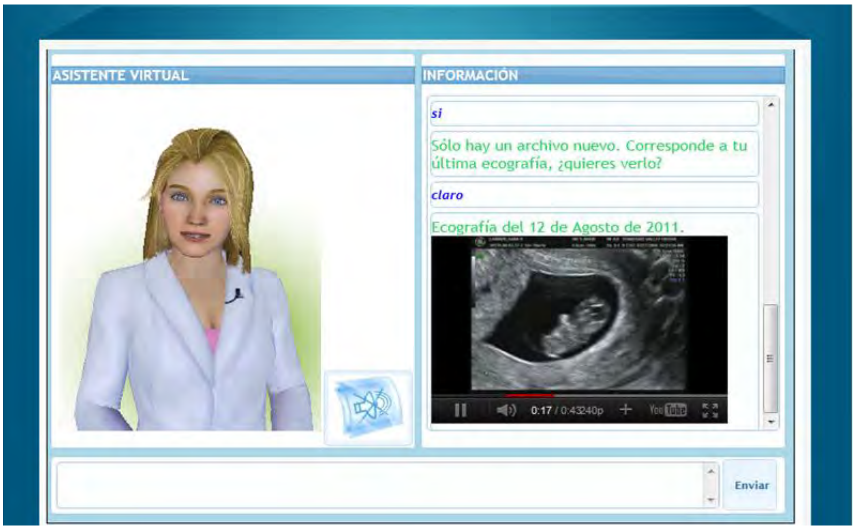
Antonio González Ruiz
Edad: 7 años

Jose González Jiménez
Edad: 5 meses

Figure 3: Expanding a headline: ECA helping with a medical report (ultrasound video)

Here we can see how the ultrasound video is presented along with the PDF link to the doctor‟s report. The avatar here is to read the report to the user so she/he doesn‟t need to open the report and read her/himself. Here we can see an instance of an ECA with a short presence: if we click again on the headline of the menu item, the avatar collapses along with the rest of the content.

In addition to helping with these reports, the avatar has a more interactive and presence-intensive place in the „Help‟‟menu, accessible via the menu bar at the top of the screen as shown in Figure 4.



si

Sólo hay un archivo nuevo. Corresponde a tu última ecografía, ¿quieres verlo?

claro

Ecografía del 12 de Agosto de 2011.

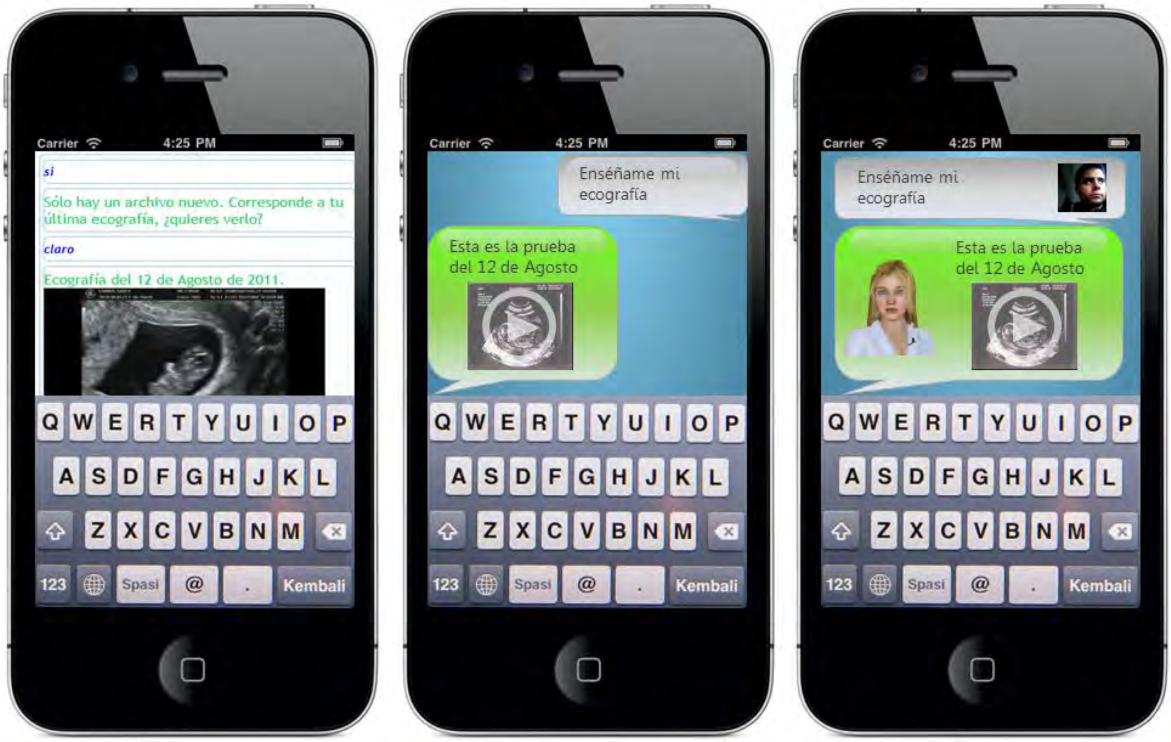
Figure 4 Virtual Help Assistant

Figure 4 shows a screenshot of the prototype Virtual Help Assistant. Here, the ECA is interactive and able to respond to user questions using dialogue generation techniques that combine video that is explained by the avatar using pre-defined utterances and chat-bot (ELIZA[4]-like) responses to topics outside of the system to permit a maximum interactivity. As we can see, to the right of the avatar there is a chat area that allows the system to present the log of the conversation, in addition to the multimedia files such as images or videos that the user may ask for (e.g., the ultrasound scanner in Figure 4) or other media that may help the avatar to explain topics.

[4] http://en.wikipedia.org/wiki/ELIZA The ELIZAchatbot

4.1.2 Adapting HealthDrive to mobile devices

If we were to port any of those two SFEs to a mobile phone, we would soon run into problems. Let us focus on the latter UI as it is more complex. We could think that by rearranging the information elements we would achieve our aims. That proves to be a complex process, as hinted in Figure 5.



si

Sólo hay un archivo nuevo. Corresponde a tu última ecografía, ¿quieres verlo?

claro

Ecografía del 12 de Agosto de 2011.

Enséñame mi ecografía

Esta es la prueba del 12 de Agosto

Enséñame mi ecografía

Esta es la prueba del 12 de Agosto

Figure 5 Different approaches for a Virtual Help Assistant on a touchscreen smartphone: Left: Direct rendering of the desktop system. Middle: Adaptation using chatting with text messages. Right: Adaptation preserving the ECA functionality.

From left to right, three mock-up alternatives of adaptation are presented. The leftmost is a more or less direct rendering of the desktop system. In it we can see that using the same exact rendering as the website results in a very poor experience, as the chat window is designed for viewing in a larger screen. This is evident in presenting the very same video window shown at the actual viewing size. The system also almost inevitably loses the avatar, since the screen is already full with the chat window. Also very importantly, we don‟t adapt to the platform UI language, further confusing the user; as she/he is using a mobile platform, we should endeavour to present familiar UI elements.

The alternative in the centre presents the same application but adapted by using the visual metaphor often presented to users in a smartphone for chatting with text messages. There the system is improved considerably just by incorporating two ideas: the balloons for clearly separating user and system messages and the usage of an appropriate iconic representation for media content such as the video thumbnail used. Tapping on the thumbnail would open the media in its own view, temporarily hiding the rest of the UI. In this way we have greatly solved the problem with the cramped UI in the left solution.

However, if we compare with the functionality shown in the desktop version, we notice a number of missing elements, the most important of which is the representation of the avatar. This is indeed a very important feature, and its omission defeats our purpose of using ECAs to improve the user‟s trust in the system as well as establishing a link with the virtual character across platforms as stated in (Cavazza, 2010) and others. Thus we may introduce ways of visually adding the ECA to the conversation, such as the one shown in the rightmost image. There, thumbnails of the user‟s and the avatar‟s resemblance are inserted in the chat balloons to further reinforce the concept that the interaction is actually a conversation.

However this is still not an optimal solution. One of the key aspects of human-machine dialogue stressed in the literature referenced earlier in this section is that non-verbal elements of the conversation (e.g., gestures, emotions) are often as important as the content in itself. However, the UI presented in the rightmost image is a very restrictive environment for these purposes, as the image of the ECA would be at most fingerprint-sized, greatly limiting our options for displaying emotions or avatar gestures. As of the writing of this document we are investigating along with real users other, more optimal options for doing this, such as text, sounds or graphics effects (e.g., changing the face of the avatar to show emotions, using emoticons in the text or playing appropriate sound cues).

In addition to these specific requirements of this generic smartphone discussion, we will need to take into account one of the most evident development requirements: that UIs adapt for different phone models, screen sizes and capabilities. Along these lines, we will now explain here how we will also target tablet-like devices, for they present a great potential for E-Health applications and also present interest for the SERENOA project as yet another family of devices in which to run applications. These devices offer operating systems and interaction patterns close to that of smartphones, but also larger screen size, yet not the same as a desktop PC. However, the tablet version will be closer to that of the desktop PC, as elements now can be kept at a similar size, than to the smartphone, whose limitations are stricter.

But limitations in smartphone/tablet systems compared to PCs have other side than just size issues, as we have explored in this section. Software limitations in the operating systems of such platforms will be also tackled in SERENOA, in particular the case of avatar engine support. This is a technical subject whose in-depth discussion is beyond the scope of this deliverable, but now we will introduce the concepts, lay out the problems and try to provide with the solutions that will be incorporated in the SERENOA system.

The PC version of the prototype runs inside a browser, but the avatar part of the UI is a non-standard part of the browser which is run via a plug-in. This is provided by the Haptek[5] company and it is available out-of-the box as a compiled solution (e.g., no modifications to the system are possible). Currently the plug-in only supports a limited scope of underlying OSs and browsers. Tablets and smartphones are not supported. Therefore it is impossible to run the system in these platforms without limitations.

4.1.3 Solution based on the SERENOA framework

What we will propose for SERENOA is a graceful degrading of the technology depending on the context (e.g., browser and OS). The first stage of degrading will be using pre-recorded videos that portrait the ECA performing the needed interactions; using this, the ECA changes to dynamically generating animations, gestures and speech to having a list of possible options from which to select to perform the dialogue actions. This, of course, will require adaptations strategies as the scope of possibilities will be more limited than when using the „live"engine and text-to-speech systems. We expect a SERENOA avatar module to be able to perform this degrading automatically. For this we will need to clearly state the UI (in this case, the interactions) using abstractions that will be made into concrete entities by the SERENOA infrastructure. A general overview of this process is presented in Figure 6.

[5] http://www.haptek.com/ : Haptek, Inc. Website

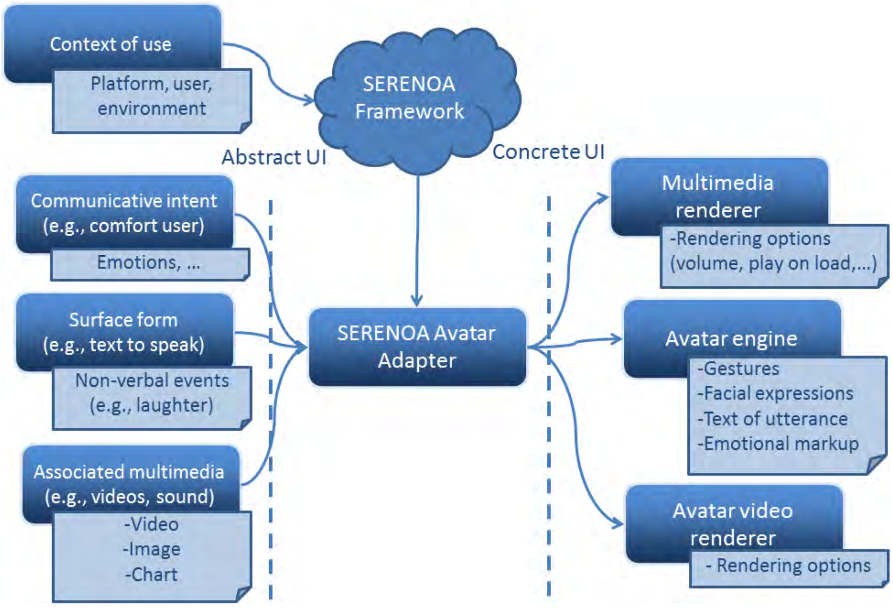


Figure 6 Dialogue adaption

As we can see, the dialogue will be generated by the application in abstract terms for three main elements:

- **the communicative intent** (e.g., the dialogue act that a given utterance will represent serve such as asking a question, trying to comfort a patient receiving some unexpected results from tests),
- **the surface form** of the dialogue act (e.g., the actual text of the utterance to be spoken plus any non-verbal cues such as gestures and non-verbal speech patterns) and
- **the associated multimedia** presented to the user (e.g., a video, a data chart or a medical image).

This content (which corresponds to the Abstract UI level of SERENOA) is then passed on to the content adapter, which is a SERENOA module that applies the needed actions to transform it into a concrete representation of the UI in terms of actions for the avatar engine, the avatar video renderer and/or the multimedia renderer. The target platform may or may not include all these three elements, and other user- or environment-related issues may affect the adaptation strategy, so the adapter needs to be informed of the context of the running system.

Summarizing, to tackle the development of our interoperable avatar system, we will leverage on SERENOA technology to

a) semantically describe our UI in design time,

b) select an appropriate rendering engine based on Context information that informs us of the user profile and the current device and

c) render the abstract interface in a way that maximizes user satisfaction.

At all times the software renderers will need to keep up with the abstract avatar behaviour and emotion generation mark-up that is done for all the versions of the application (desktop, mobile). Here, having a multi-platform environment such as SERENOA will be undoubtedly crucial. The details of the resulting prototypes and the design principles of the AUI and CUI descriptions will be further explained in future revisions of this deliverable (D5.2.2) and other documents in the SERENOA project such as D3.2.1.

4.1.4 Development process

In order to have constant user feedback for our designs, during the coming months after the writing of this document we will further subject these mock-up designs and preliminary prototypes to a panel of potential users of the technology and ECA experts to gather relevant feedback upon the needs for adaptation in an avatar-based SFE. The results of this study will be used to steer some of the developments in SERENOA to provide the elements in the infrastructure needed to power these prototypes by the end of the project.

4.2 E-Commerce scenario by W4

As such, W4‟s E-Commerce scenario does not correspond to any real-life business case. Instead, it gathers, within a single application, requirements inspired by various customers of W4 who have expressed interest in the outcomes of the SERENOA project, and who would like to see some impact in W4‟s product line. Such customers have already implemented business applications using W4‟s products and have asked, in the past, extensions to their existing systems in order to meet new requirements that have to do with context aware adaptation of Graphical User Interfaces. These customers notably include:

- A public French administration who want to make its information portal compatible with accessibility standards to allow people of all abilities and disabilities
- A software provider, specialized in the field Business Intelligence (BI), operating in Canada, where end-users are likely to request service either in French or English
- A middleware provider, selling secured payment solutions to E-Commerce actors, who would like to make its systems available on mobile devices, namely smart phones and tablet PCs.

These high-level requirements have in common that they require SFE adaptation based on different criteria. W4 therefore imagined a prototype where system adaptation could be tackled and came up with a sample E-Commerce scenario, where SFE adaptation cases will be explored, including adaptation to:

- language
- device: desktop, smart phone or tablet PC
- users‟‟disabilities, such as eye impaired or color blind persons

The intent of the scenario was to create a demonstrator that can illustrate a scenario where different end-users, either on the back-end side (the seller, or merchant web site) or the front-end side (shoppers) collaborate to conduct a successful transaction.

The typical workflow will thus sequentially involve the following factors:

- An online shopper, who needs device adaptation (either a home computer either a mobile device) or accessibility features (such as low visual perceptibility), and who will purchase a list of items;
- An employee working for the merchant site, either in French or English, who also needs accessibility features (color-blind), who will insure service by being in charge of the order. This includes:
    - Making sure the order is valid
    - Charging the customer and holding back the money until the item is successfully delivered
    - Finalizing the transaction by paying the merchant site once delivery has occurred

4.2.1 Current status of the demonstrator

The E-Commerce prototype needs to be developed from scratch, following the steps described in Section 4.2.3 and it is only in its early phase. However, W4‟s SERENOA team has already obtained interesting results in terms of adaptation of existing web applications to mobile applications, which is the first type of adaptation explored in the context of this scenario.

4.2.2 Adapting existing applications to mobile devices

The adaptation of existing applications to mobile devices is an adaptation field which is currently explored by W4‟s R&D. The first chosen mobile platform for this work is Android.

For this purpose, W4 has identified a subset of its graphical widgets where generic mechanisms need to adapt the GUI. Sample applications were used to adapt:

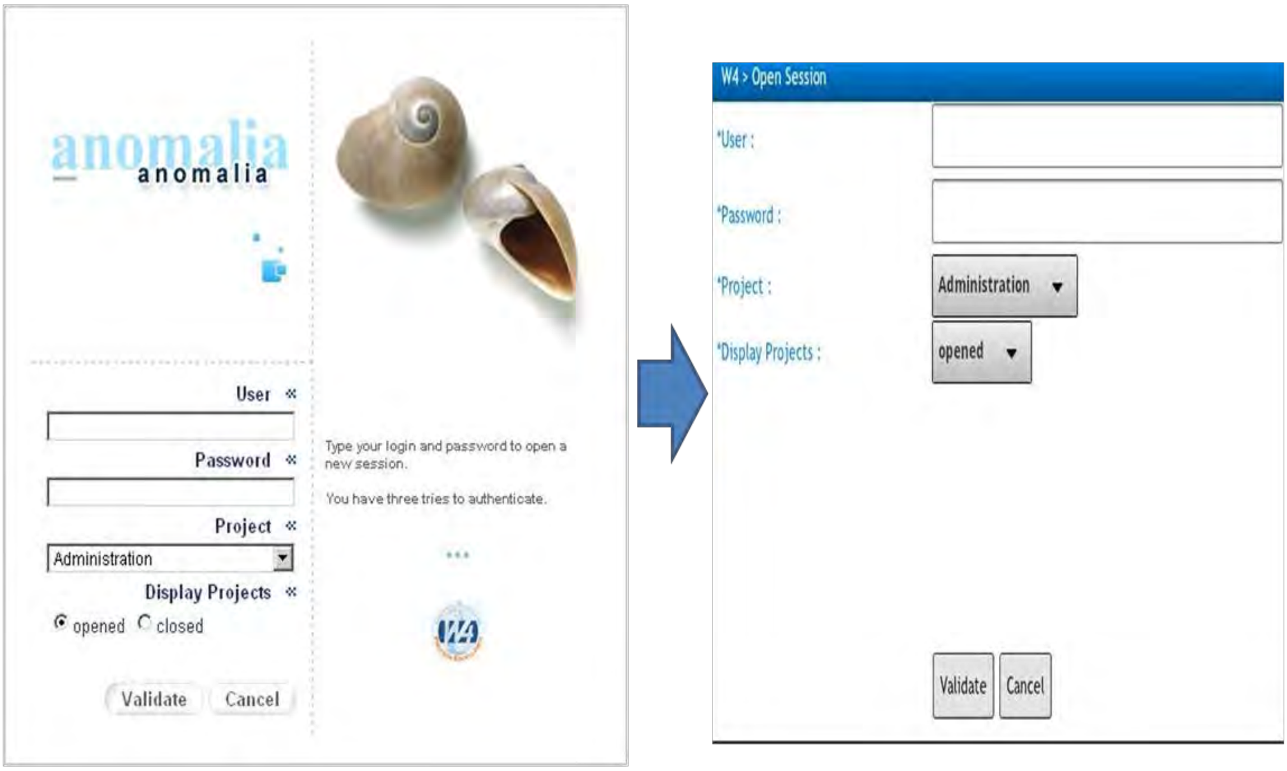
- Table view ,
- Forms (for CRUD operations)
- Chart views

W4 relies on a model interpretation engine (called ***Application Engine***) to generate GUI components directly from the model. The most interesting way for adapting applications is therefore to reuse their model in the following way:

1. Process dynamically the views generated by this engine (from which the output is a screen description described in XML) and adapt them to the mobile device. The views are parsed by the native platform (Android in this case) and then transformed in order to offer ergonomics compatible with the used terminal. The views are parsed using a SAX parser and to build an Android native widget hierarchy defining the structure of the views to display.
2. Send back the user‟s request for form validation (example: add a new row in a table) or for navigating to another screen to the engine for further processing.

W4‟s R&D used a native Android application to implement adaptation rules, with property files that are used for customizing our mobile applications and to define the colour, the font and the size of each graphical element. Generic XML files are also designed to define the structure of generic graphical components (for example, a field is always composed of 2 or 3 parts, starting with a label.

Below are some examples of the results obtained, showing some results on typical windows. These adaptation choices from desktop to the Android platform will be enhanced as the SERENOA project unfolds. The left screenshot represents the existing web page of the application and the right screenshot the current result in terms of adaptation for the Android platform.



User «

Password «

Project « Administration

Display Projects « ● opened ○ closed

Type your login and password to open a new session.

You have three tries to authenticate.

W4 > Open Session

*User :

*Password :

*Project : Administration ▾

*Display Projects : opened ▾

Figure 7: Login screen adaptation, with transformation of check boxes to combo boxes. Graphical skins still need to be adapted.

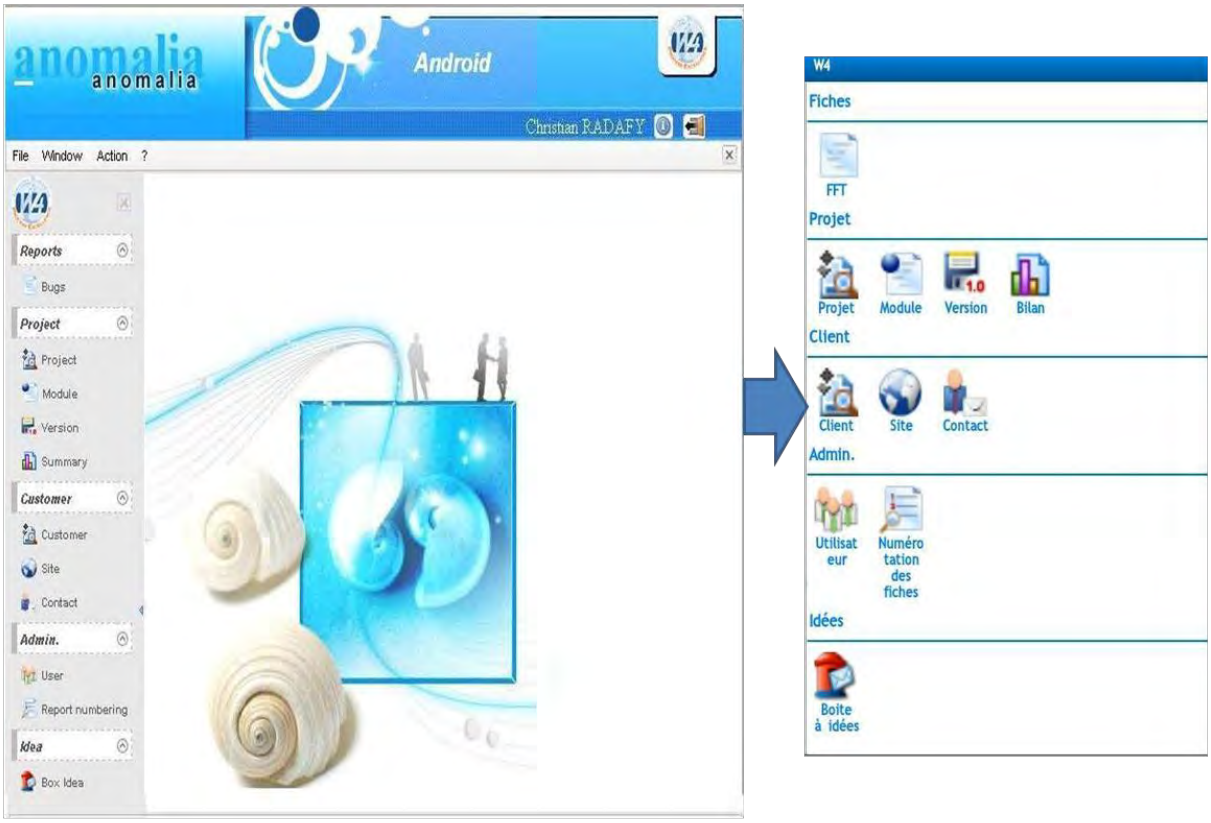
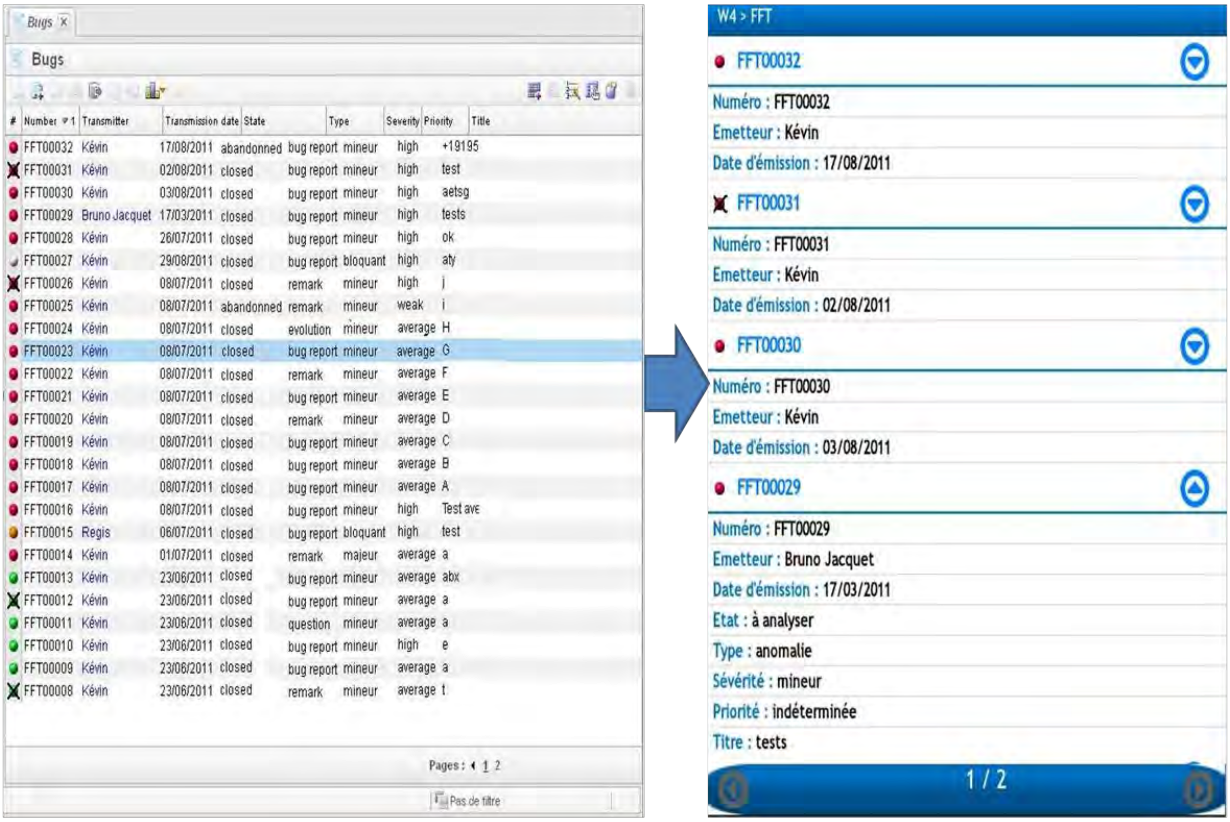


Figure 8: Menu adaptation, where groups of options (expandable tabs) and menu options are laid out using expandable tabs to action groups to optimize space use



#	Number ▼1	Transmitter	Transmission date	State	Type	Severity	Priority	Title
●	FFT00032	Kévin	17/08/2011	abandonned	bug report	mineur	high	+19195
✗	FFT00031	Kévin	02/08/2011	closed	bug report	mineur	high	test
●	FFT00030	Kévin	03/08/2011	closed	bug report	mineur	high	aetsg
●	FFT00029	Bruno Jacquet	17/03/2011	closed	bug report	mineur	high	tests
●	FFT00028	Kévin	26/07/2011	closed	bug report	mineur	high	ok
●	FFT00027	Kévin	29/08/2011	closed	bug report	bloquant	high	aty
✗	FFT00026	Kévin	08/07/2011	closed	remark	mineur	high	j
●	FFT00025	Kévin	08/07/2011	abandonned	remark	mineur	weak	i
●	FFT00024	Kévin	08/07/2011	closed	evolution	mineur	average	H
●	FFT00023	Kévin	08/07/2011	closed	bug report	mineur	average	G
●	FFT00022	Kévin	08/07/2011	closed	remark	mineur	average	F
●	FFT00021	Kévin	08/07/2011	closed	bug report	mineur	average	E
●	FFT00020	Kévin	08/07/2011	closed	remark	mineur	average	D
●	FFT00019	Kévin	08/07/2011	closed	bug report	mineur	average	C
●	FFT00018	Kévin	08/07/2011	closed	bug report	mineur	average	B
●	FFT00017	Kévin	08/07/2011	closed	bug report	mineur	average	A
●	FFT00016	Kévin	08/07/2011	closed	bug report	mineur	high	Test ave
●	FFT00015	Regis	06/07/2011	closed	bug report	bloquant	high	test
●	FFT00014	Kévin	01/07/2011	closed	remark	majeur	average	a
●	FFT00013	Kévin	23/06/2011	closed	bug report	mineur	average	abx
✗	FFT00012	Kévin	23/06/2011	closed	bug report	mineur	average	a
●	FFT00011	Kévin	23/06/2011	closed	question	mineur	average	a
●	FFT00010	Kévin	23/06/2011	closed	bug report	mineur	high	e
●	FFT00009	Kévin	23/06/2011	closed	bug report	mineur	average	a
✗	FFT00008	Kévin	23/06/2011	closed	remark	mineur	average	t

● FFT00032

Numéro : FFT00032

Emetteur : Kévin

Date d'émission : 17/08/2011

✗ FFT00031

Numéro : FFT00031

Emetteur : Kévin

Date d'émission : 02/08/2011

● FFT00030

Numéro : FFT00030

Emetteur : Kévin

Date d'émission : 03/08/2011

● FFT00029

Numéro : FFT00029

Emetteur : Bruno Jacquet

Date d'émission : 17/03/2011

Etat : à analyser

Type : anomalie

Sévérité : mineur

Priorité : indéterminée

Titre : tests

Figure 9: Table view adaptation, where table rows are displayed on multiple lines with expand/shrink features

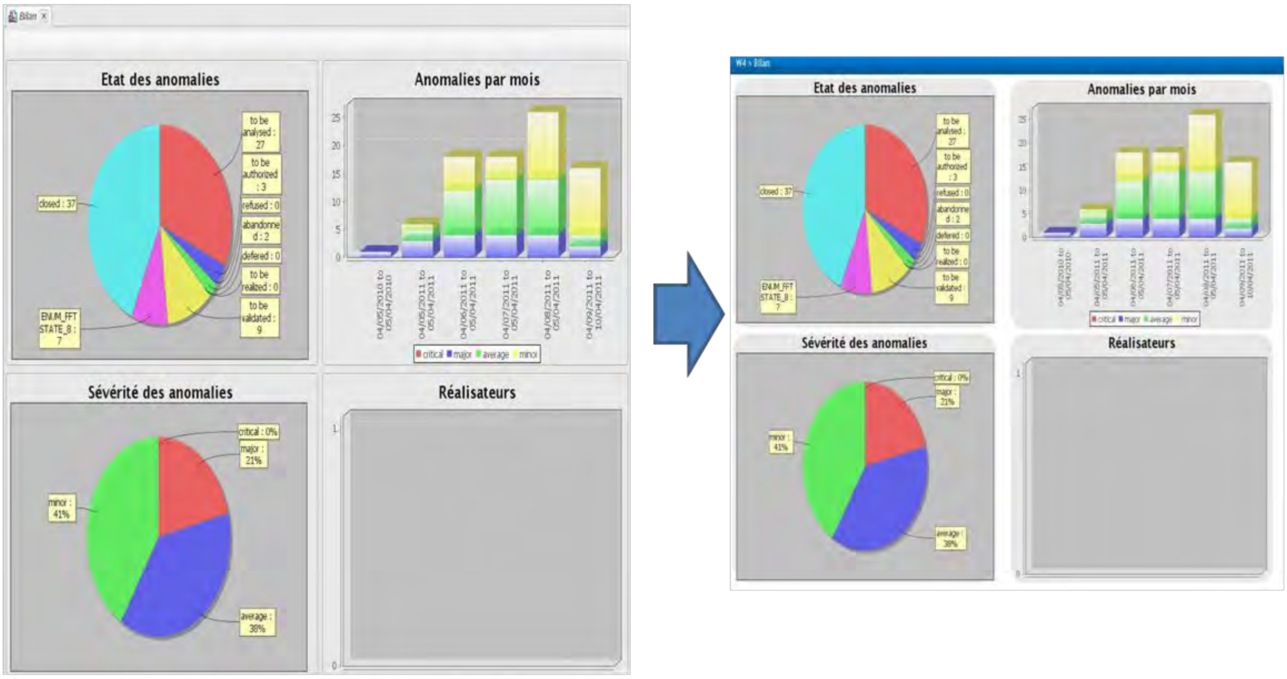
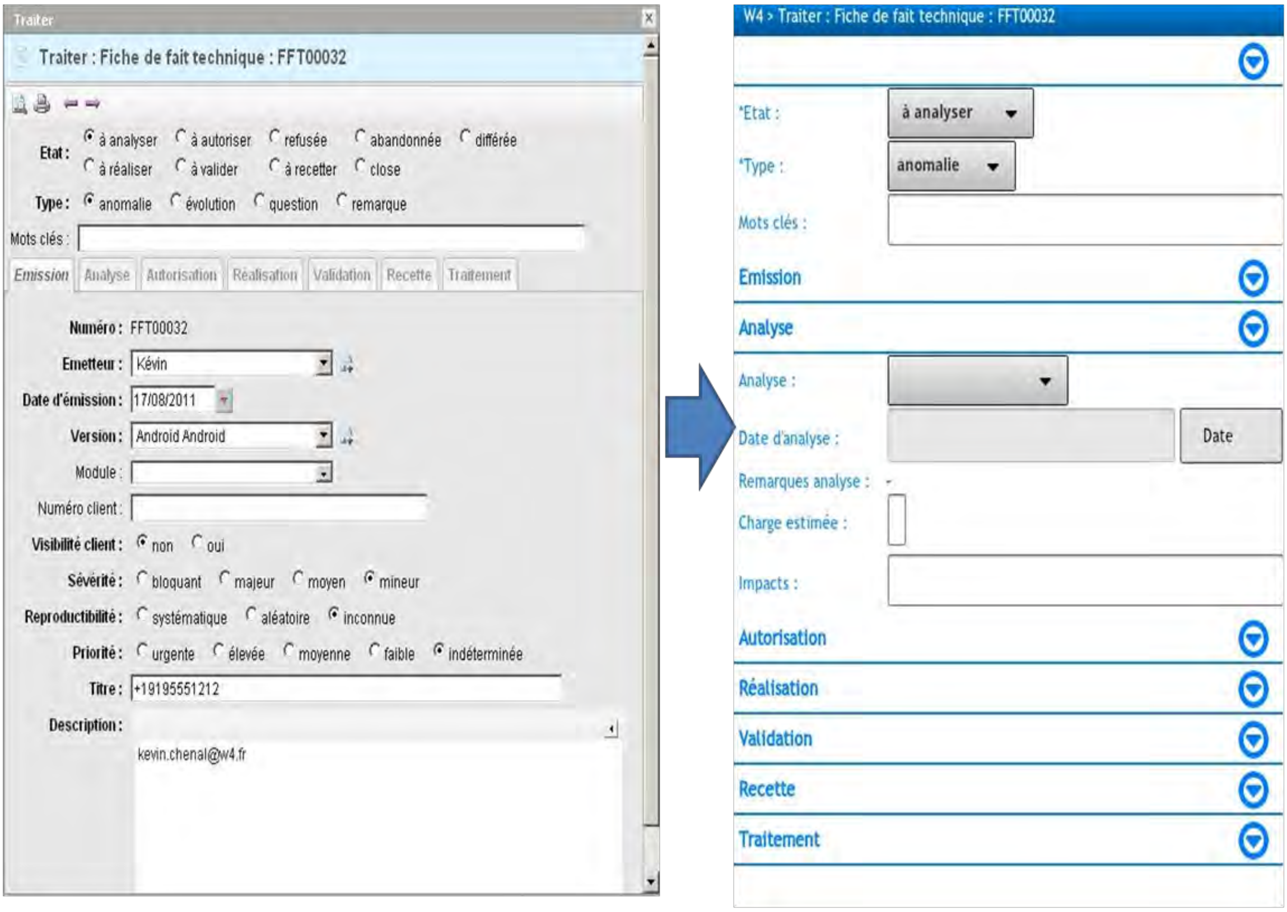


Figure 10: Chart adaptation: for now, the initial page is simply transformed in an image and sent to the mobile


Traiter

Traiter : Fiche de fait technique : FFT00032

Etat : ● à analyser ○ à autoriser ○ refusée ○ abandonnée ○ différée ○ à réaliser ○ à valider ○ à recetter ○ close

Type : ● anomalie ○ évolution ○ question ○ remarque

Mots clés :

Emission | Analyse | Autorisation | Réalisation | Validation | Recette | Traitement

Numéro : FFT00032

Emetteur : Kévin

Date d'émission : 17/08/2011

Version : Android Android

Module :

Numéro client :

Visibilité client : ● non ○ oui

Sévérité : ○ bloquant ○ majeur ○ moyen ● mineur

Reproductibilité : ○ systématique ○ aléatoire ● inconnue

Priorité : ○ urgente ○ élevée ○ moyenne ○ faible ● indéterminée

Titre : +19195551212

Description : kevin.chenal@w4.fr

W4 › Traiter : Fiche de fait technique : FFT00032

*Etat : à analyser

*Type : anomalie

Mots clés :

Emission

Analyse

Analyse :

Date d'analyse : Date

Remarques analyse :

Charge estimée :

Impacts :

Autorisation

Réalisation

Validation

Recette

Traitement

Figure 11: : Form adaptation: Check boxes are replaced with combo boxes, tabs are replaced by expandable areas, native calendar components replace Ajax components.

The Android development toolkit offers prebuilt multiple views, including widgets and layouts that you can use to build the UI. Used components include Button, TextView, EditView, ListView, CheckBox, RadioButton, Gallery, Spinner, and the more purpose specific AutoCompleteTextView and TextSwitcher. The used layouts are LinearLayout, FrameLayout, RelativeLayout.

4.2.3 Development process

The W4-team use, for their development process, a combination of agile and RAD methodologies which has been presented in deliverable D3.4.1. Using the LEONARDI model-driven software suite, the W4 developers iterate cycles based on the following steps:

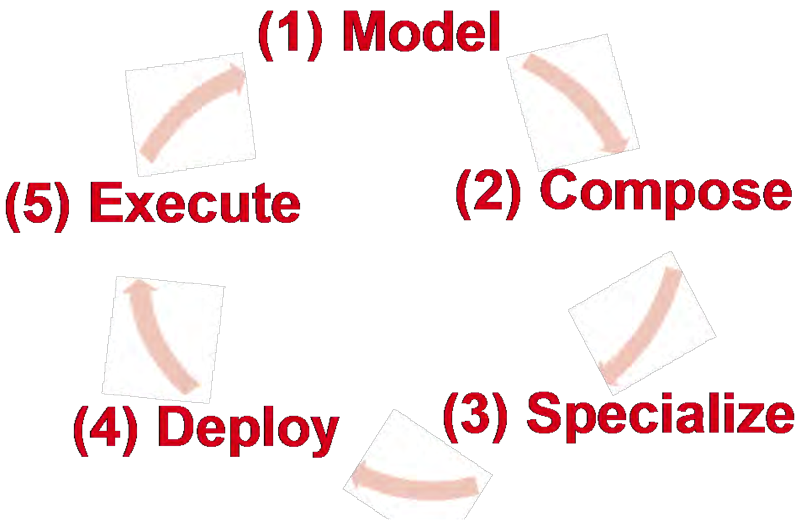


Figure 12: RAD cycles at W4

1. **Data modelling:** This is where most of the work is done. For the authoring tool, this step consists in defining the data structure for creating and managing adaptation rules. Classes, fields and relationships between classes are defined in the model. For the scenario, the data model will reflect the entities that are needed for implementing the E-Commerce scenario.
2. **GUI composition and configuration**: From the output of step 1 (the business model), the Leonardi engine automatically renders, at execution, the Graphical User Interface. Thus, the screens for creating, editing and removing new adaptation rules are automatically available with ready-to-use services, such as sorting, filtering or printing. During this development step, the GUI is specialized to fit in a more appropriate way the needs of the application (SERENOA authoring tool). The form layouts are modified, value formatting is defined, as well as navigation in the application. This step usually does not require coding, but only configuration in the design environment.
3. **Behaviour specialization**: All behaviours inherited from Leonardi are customizable. Behaviours are application related tasks triggered by events occurring as the final user interacts with the application: logging in, clicking a menu option, validating a screen, quitting the application... Adding customized rules when such events occur is made possible by adding Java code that will be called on the fly at execution time by the Leonardi engine. This step is optional when early version of the prototype are developed, which increases the RAD performance. Usually, 80% of the functional needs can be covered by Leonardi"s generic features.
4. **Deployment**: Once the model and, possibly, specialized code are available, they are embedded together with the Leonardi engine in an archive (typically a WAR[6] file) and deployed to the target platform.
5. **Execution**: When the application is started, the Leonardi engine dynamically executes the model,

[6] http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/WCC3.html : Web Archive format

producing the screens as requested by the end user and calling the specialized code when needed

Initially, only a model describing the business model is needed. Therefore, such cycles can be very rapid. At the end of each cycle, a new, enriched version of the prototype is delivered that can be shown to the project stakeholders (the SERENOA partners, end-users) for additional feedback that feeds a new cycle.

W4"s main business is to develop core business applications deployed on the web, using its own Java-based products. In this context, W4 usually uses a model-driven approach and an agile, Scrum-like project methodology to iterate on various versions of the software, from early prototypes to the final application. MDE (Model Driven Engineering) is W4"s traditional way of implementing business solutions for its customers. Besides, application adaptation to mobile devices is a short-term technical goal for W4.

Reusing this know-how and applying the same development methodologies, the natural 3 step approach chosen by W4 for implementing the sample E-Commerce scenario consists of:

(1) **Upstream R&D work for adapting existing applications.** One of the key aspects of adaptation for W4 is that it needs to be applicable to existing applications at low cost. This constraint stems out from W4"s business context, where customers are explicitly asking to upgrade their existing solutions to make them adaptable to various contexts of use. The first adaptation field explored by W4"s R&D is adaptation of existing applications to mobile devices. Implementation of the E-Commerce application.

(2) **Implementation of the E-Commerce application**, as described in SERENOA. This implementation will be done in a traditional way by W4, applying the MDE approach as well as an agile project methodology. The result will then serve as a sample application for adaptation purposes. The first task (currently in progress) for implementing the E-Commerce application based on the W4 product LEONARDI is to develop a domain model, where all the business classes are represented with their attributes: users, user profiles, items etc. This model will then be executed by the model interpretation engine to implement both the back-end context (for the seller) and the front-end context (for the online shopper), thus offering access to all the types of end-users identified in the E-Commerce scenario.

(3) **Adaptation of the E-Commerce application** developed during step (2) by applying work done during step (1). In an ideal world, this step should occur automatically and only used to visualize the adaptation effects of the various CAA mechanisms resulting from SERENOA.

4.3 Intelligent Picking Prototype by SAP

As described section 2.3 the SERENOA prototype will enhance the demonstrator for intelligent picking at the Future Retail Centre in Regensdorf, Switzerland.

In the early age of wearable computing, researchers focused on the tasks of inspection, repair and maintenance as the potential areas of its application (Smailagic et al., 1998 and Siewiorek et al., 1998). Siewiorek et al., (2008) provided an overview of the lessons learned from user studies of deployed prototypes in these areas. Commercially, one early success by Symbol Technologies (acquired by Motorola) was in creating an arm-mounted barcode scanner that could speed package scanning and inventory control (Stein et al., 1998). Pittsburgh-based Vocollect addressed another niche, inventory picking, using their speech-only interface (Starner, 2002). While Vocollect has been successful, the user studies mentioned above for similar problems suggest that head-mounted displays (HMD) might also prove useful for the task of inventory picking.

Picking is the process of collecting items from an assortment in inventory and represents one of the main activities performed in warehouses. It accounts for 55% to 65% (Bartholdi and Hackmann, 2009 and Coyle et al., 2002) of the total operational costs of a warehouse. Typically the process begins with a picking list, which specifies the location of each type of item, the number of items to be picked, and the sequence in which the items will be picked. A worker collects the items from stock and transports the items to a specific location.

The SERENOA target environment is one where a wearable computer interface can demonstrate improvements over traditional methods, and the metrics used to evaluate the system have sufficient sensitivity to show the effect of changes to the interface. Based on the past commercial successes, the picking seems a logical choice, because it has the benefits of being simple to teach to naive users and fast to perform so that many trials may be executed in a small period of time. Hopefully, the learning effects can be modelled and expert users can be trained quickly, if desired. In addition, picking can be made mentally, visually, and manually taxing, similar to many other tasks that are being investigated for augmentation with wearable computers. The standard quantitative metrics, such as performance time and accuracy, and subjective metrics, like the NASA-TLX (Task Load Index), can be applied to experiments. Finally, picking also has a simple, ecologically-valid control condition that can be used for comparing the paper based picking list.

On the other hand, distributed programming paradigms have emerged in recent years that allow generic software components to be developed and shared. Although ideal for some enterprise integration and E-Commerce, it has only been with the adoption of XML as common data syntax that the underlying principles have gained wide scale adoption through the definition of Web Service standards. According to the definition of W3C, "A Web Service is a software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts" (Cabral, 2004). The Web Services are well defined, reusable, software components that perform specific, encapsulated tasks via standardized Web-oriented mechanisms. They can be discovered, invoked, and the composition of several services can be choreographed, using well defined workflow modelling frameworks. A Web Service supports direct interactions with other software applications (i.e. agents) using XML based messages exchanged via internet based protocols.

Wearable devices have some properties which affect the design of the User Interface. First of all, wearable devices on HMD technology have usually a small screen resolution (800*600). Secondly, wearable devices very often follow the “hands free” paradigm. Instead of using mouse or keyboard other interaction concepts like gestures and speech are used. Systems which are based on wearable devices, very often take advantage of environment which are equipped with special sensors like Laser Range Finder and RFID tags. This connection to a smart environment also affects the User Interface. Finally, the task which needs to be accomplished by the user is strongly related to the specialized component of Enterprise Resource Planning (ERP) System to which the device needs to be connected. A relationship between the model based UI and the Data model from the backend needs to be established.

The SERENOA prototype will be built upon these wearable and mobile technologies, smart environments and the data from ERP systems. The goal is a UI which can be adapted to this context applied to a logistics task. In the remainder of this section we will summarily describe the actual state of the prototype and some ideas for the future development that will be used in SERENOA.

4.3.1 The current architecture of the system

This section describes the current status of the picking system. It explains the components of External Enterprise Resource Planning (ERP) System, Middleware, Context Providers, Client, SAP Supply Chain Management (SCM) System and Security. In the following sections, each component is described briefly.

4.3.1.1 External ERP System (EERPS)

The EERPS server is a backend system, integrated with a relational database system, which stores the orders‟ information, which can be exported in the form of CSV (Comma Separated Values) data files. Each file contains information about orders, with a list of items to be picked by the picker, including their names, locations (shelf, box), and quantities. These CSV files are transferred via FTP to the middleware, or can also be copied manually at a particular location or directory of it.

4.3.1.2 Context Providers

The context providers are used to provide raw contextual data to the Context Server, where it is further processed to be used for the overall context awareness. In the following sub-sections, the functionality of each context provider is described briefly.

The Laser Range Finder (LRF), mounted in the middle on top of a shelf, provides the interactional context data, which means the data about the picker‟s hand/arm interaction with the shelf and its boxes during the picking of required items. The data obtained from the LRF are the distances to the picker‟s hand at defined angles when intercepted by the LRF‟s invisible laser curtain, hanging from the top of the shelf to its bottom. These distances are used to determine a position with respect to the known geometry of the shelves to determine which box, if any, was used for picking. This information is further processed with statistical evaluations and heuristic algorithms to enable the tracking of individuals‟ hands. By using the tracking information, the context information can be provided to the system in the form of „entered in box X‟, „left box X‟and „moved from box X to box Y‟.

The Weight Scale, placed in the shelf under each box, provides the quantity of items that have been picked by the user from this particular box. This quantity is then compared with the desired number of items that was originally planned to be picked from this box in order to compute and graphically show the user whether he has picked all the necessary items or not. Furthermore, this information is sent to the backend SAP SCM system via Context Server in order to manage the inventory and demand-supply of items.

The wearable computer is used to fetch and process the interactional context data from the server and show it to the user through the attached HMD. For example, when the user picks the parts from a correct/desired box, that box is highlighted with the colour of the row, i.e. orange or yellow, as shown in Figure 13 b).

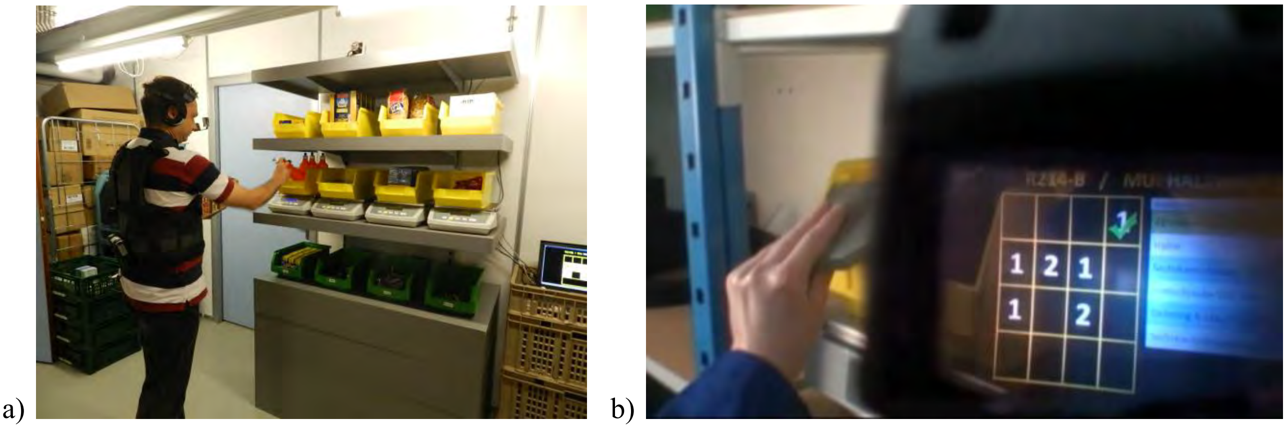


Figure 13: a) Picking with a wearable computer from a shelf at the Future Retail Center in Regensdorf, Switzerland. b) GUI seen by the User (picker) in the Head Mounted Display

4.3.1.3 Middleware

The middleware is an integral part of picking system and acts as a gateway or interface for the outer world. It receives orders‟ information from the EERPS as CSV data files, and then feeds them to the Orders/Tasks Data Server, which creates orders and tasks objects from this information, and then provides them to the clients via WSO (Web Service for Orders) and WST (Web Service for Tasks) components, respectively. Each Task comprises a list of Steps and each step comprises a list of Items to be picked by the picker. The Web Services based approach facilitates not only Java based clients (running Java Virtual Machines) but also .NET based clients (running Microsoft Windows Mobile and .NET compact framework). Secondly, the RMI Server component is developed to facilitate only the Java based clients to have relatively faster transmission of contextual information. The other important component of middleware is the Context Server. The context server converts the Polar coordinates provided by the LRF to Cartesian coordinates to compute the correct box number for the picker to avoid mistakes. The context server allows different Context Clients to register/subscribe for particular context(s) information which they are interested in, i.e. Artificial (for the computed information, e.g. application messages), Environment (information from the environment, e.g. time, temperature), Implicit (the information derived implicitly, e.g. light threshold reached), User (information from the user, e.g. button pressed), and Virtual (virtual information, e.g. free space on a media server). Besides this context information, the context server allows the user to define more contexts and their properties according to the requirements of the application scenario.

4.3.1.4 Java/.NET Client

The client is the wearable component, which the picker uses for the picking. It is equipped with the HMD (Head Mounted Display), which is used to show the Graphical User Interface (GUI) to the picker, which he uses to see the items (names, quantity) to be picked from a particular shelf and from a particular box. Also, the client comprises the context client which is used to register/subscribe on the context server for the desired contextual information.

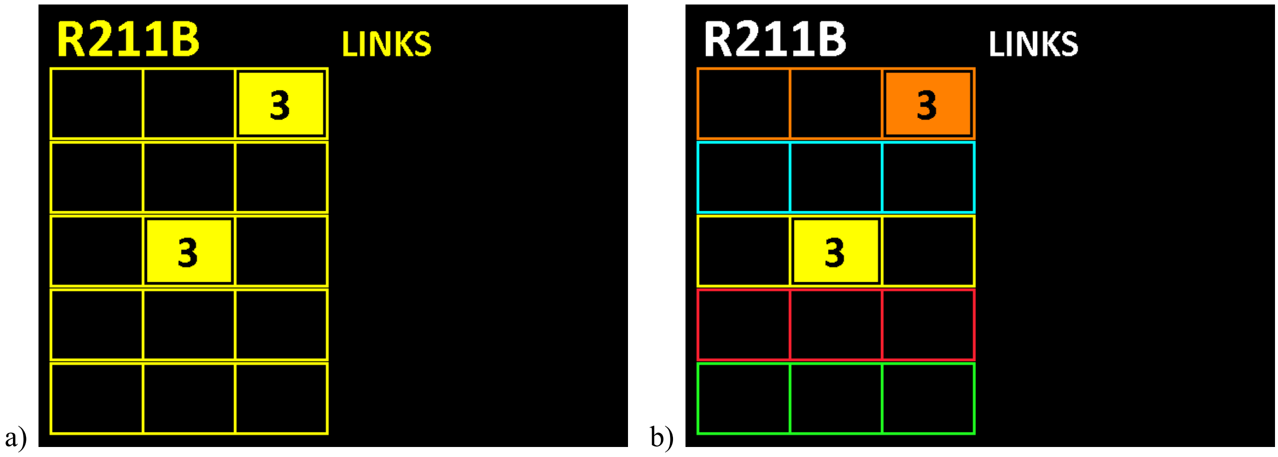
4.3.1.5 Security

The whole system is secured using different levels of security using standard mechanisms. The first and basic level is that the user uses his username/password in order to login to the Windows OS of wearable computer (Embedded XP), as well as to the server (Windows Vista). Since both of the computers are part of a secured network, the inherent network security is provided by default to the network traffic. is The second and higher level of security is that the client (running on wearable computer) and server applications exchange messages using encryption and decryption techniques as specified in WS-Security specification. The third and highest level of security, which is the extension of second level of security, additionally supports the application to use signing and verification techniques according to the WS-Security specification.

The security system is completely configurable and flexible; which means that the user can easily enable/disable different levels of security. For example, if only the second level of security is enabled, then the first level is automatically enabled, but at the same time, keeping the third level disabled.

4.3.2 User Interaction and User Design

Wearable computers require a certain manual adaptation of the screen appearance and navigation. The requirements from the industrial environment concerning interaction modalities and technologies need to be met. As described in Ali et al., (2011) one solution is the attempt to select a good UI design from a certain set of UI variants through a user study. Figure 14 shows some examples of such UI variants.



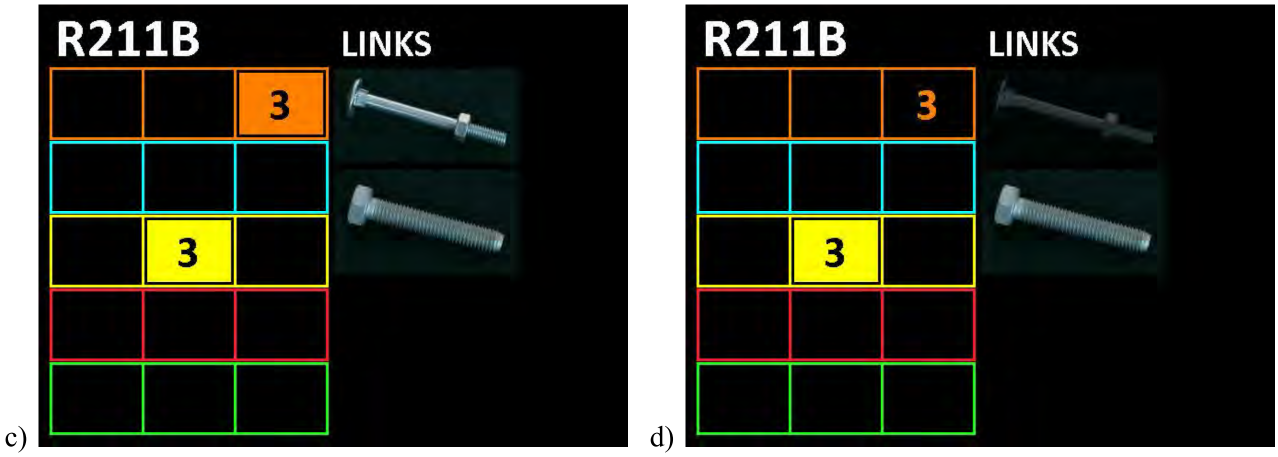


Figure 14: Different UI variants for picking: The identifier of the shelf (R211B LINKS) shown on the top of the UIs, the shelf layout is represented as a grid and the items to be picked with their amount as a number. a) monochrome, b) coloured, referring to similar coloured rows of the shelf, c) with image of the items to be picked, d) with feedback from a LRF (items have been picked).

In contrast to a “best fit” approach the paradigm of adaptive UI in SERENOA would interpret the variants as being dependent from the context. The user can choose to have an image of the item, the device can trigger the use of a monochrome display and the environment can suggest the support through a smart sensor.

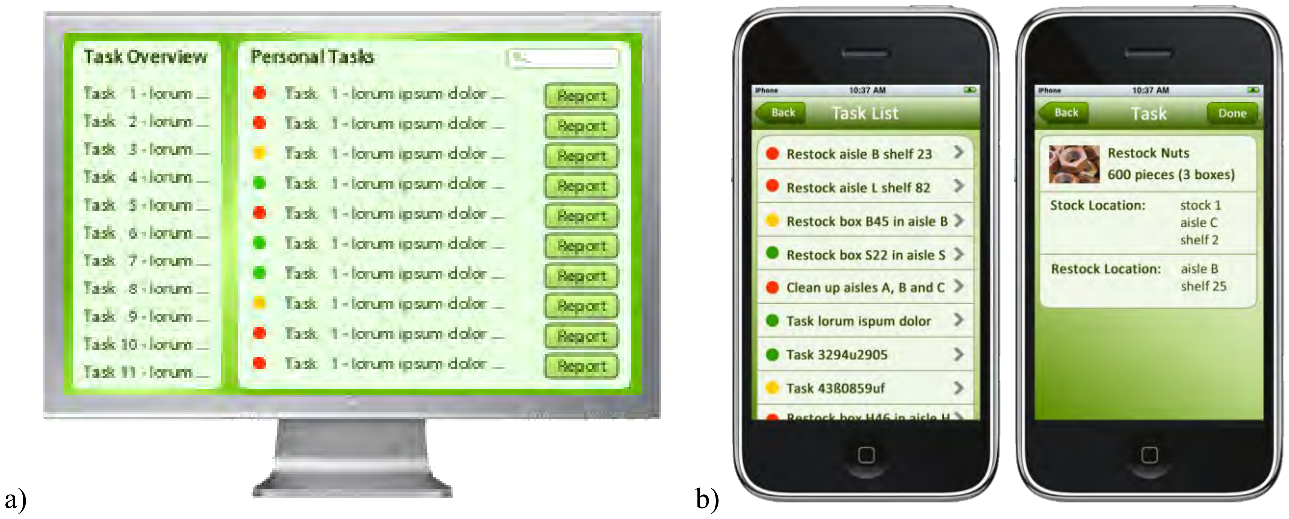
4.3.3 Adaptation to the context

In the current state of the prototype, the picker stands directly in front of the shelf for the picking, but in future, it will be leveraged with the advanced navigational and interactional capabilities by attaching RFID tags with the shelves and their boxes, so that the picker could be guided with the shortest path from his standing position to the desired shelf in large warehouses in order to save time.

The current client is GUI based with the fixed font size. However, the users having a weak eye sight want the GUI with bigger fonts; hence in future, the GUI will be developed in a way so that it could adapt according to the preferences of the user.

In order to complete an order task, the user currently uses two buttons (forward, reverse) input device in order to navigate between the steps. However, two more interactional modalities will be developed for the future version, namely gesture and speech-based. In noisy environments, gestures based interaction is definitely advantageous.

Concerning the adaptation to the platform we are currently targeting three devices, a stationary (desktop) PC, the above mentioned wearable computer and a common mobile device (Smartphone or iPad). The GUIs which hold the information about the list of orders to be processed by the picker are shown in Figure 15.


a)

Task Overview

Task 1 - lorum ...
Task 2 - lorum ...
Task 3 - lorum ...
Task 4 - lorum ...
Task 5 - lorum ...
Task 6 - lorum ...
Task 7 - lorum ...
Task 8 - lorum ...
Task 9 - lorum ...
Task 10 - lorum ...
Task 11 - lorum ...

Personal Tasks

● Task 1 - lorum ipsum dolor ... Report
● Task 1 - lorum ipsum dolor ... Report
● Task 1 - lorum ipsum dolor ... Report
● Task 1 - lorum ipsum dolor ... Report
● Task 1 - lorum ipsum dolor ... Report
● Task 1 - lorum ipsum dolor ... Report
● Task 1 - lorum ipsum dolor ... Report
● Task 1 - lorum ipsum dolor ... Report
● Task 1 - lorum ipsum dolor ... Report
● Task 1 - lorum ipsum dolor ... Report

b)

iPhone 10:37 AM
Back Task List

● Restock aisle B shelf 23 >
● Restock aisle L shelf 82 >
● Restock box B45 in aisle B >
● Restock box S22 in aisle S >
● Clean up aisles A, B and C >
● Task lorum ispum dolor >
● Task 3294u2905 >
● Task 4380859uf >
● Restock box H46 in aisle H >

iPhone 10:37 AM
Back Task Done

Restock Nuts
600 pieces (3 boxes)

Stock Location: stock 1
aisle C
shelf 2

Restock Location: aisle B
shelf 25

Figure 15: Adaptation of a graphical user interface to different platforms The devices show a list of orders which can be assigned and scheduled. a) The stationary device offers the possibility to write reports.

The third platform, the wearable device has a UI which is designed for an efficient processing of the already assigned orders.

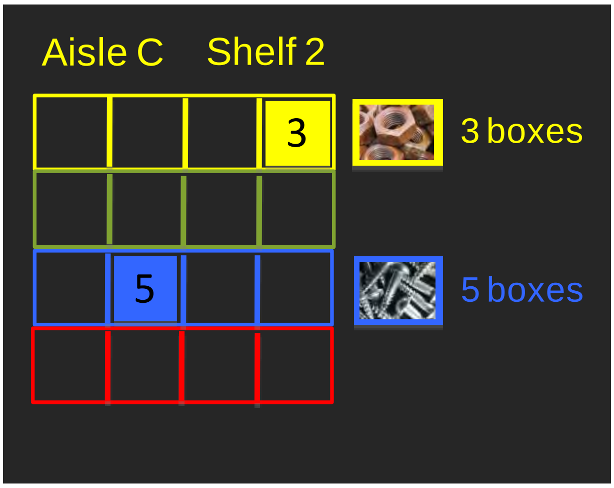


Figure 16: Graphical user interface for the wearable computer to directly support the task of picking.

The graphical user interface for the wearable computer, as shown in Figure 16, remains with a focus on simplicity. The important aspects of adaptation are the user profile, e.g. images of the items are desired, the environment, e.g. the shelves have a colour code or the shelves have intelligent sensors and the features of the platform, e.g. a speech and a gesture interface are available.

The adaptation of the UI to the user profile due to the preferred language, the level of expertise and the knowledge about the location are shown in Figure 16.

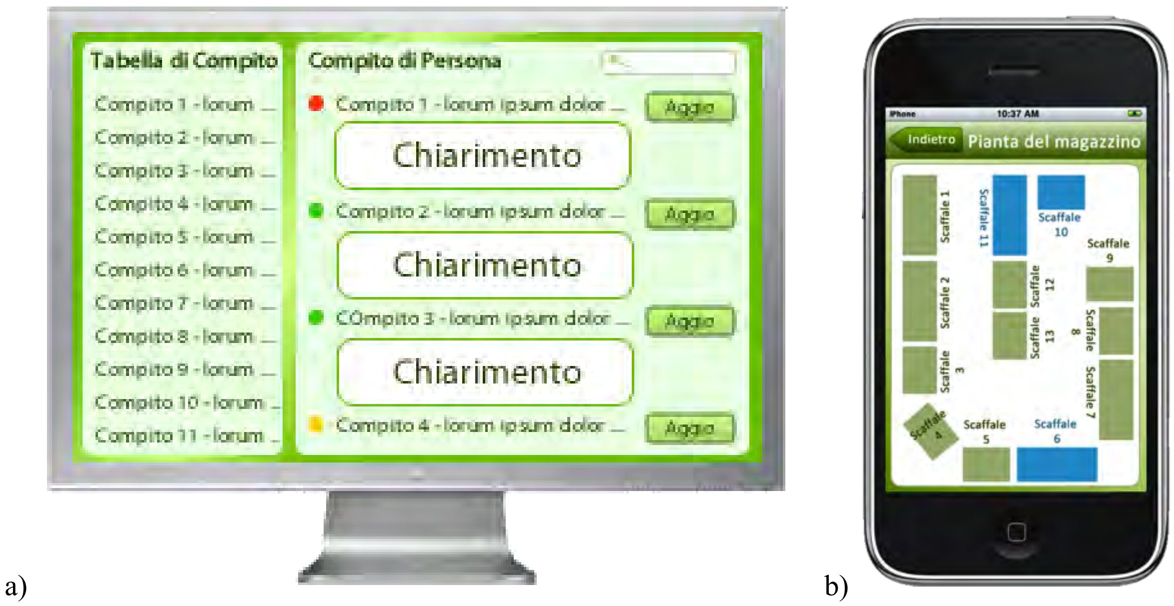


Figure 17: Graphical user interface adapting to the user profile: a) The list of orders adapts to the preferred language and to the level of expertise, adding additional help buttons. b) The UI offers an additional navigation functionality as the user does not know the location.

For the navigation with a wearable computer a UI a real-time navigation, as known from automobiles can be developed.

4.3.4 Development process

The SAP-team follows an agile methodology as described in deliverable D3.4.1

The SAP team is using different approaches when designing their prototypes. They follow an agile methodology (as described in deliverable D3.4.1) using methods of User-Centered Design (UCD) (Figure 18), which ensure iterative working style and rapid prototyping at same time.

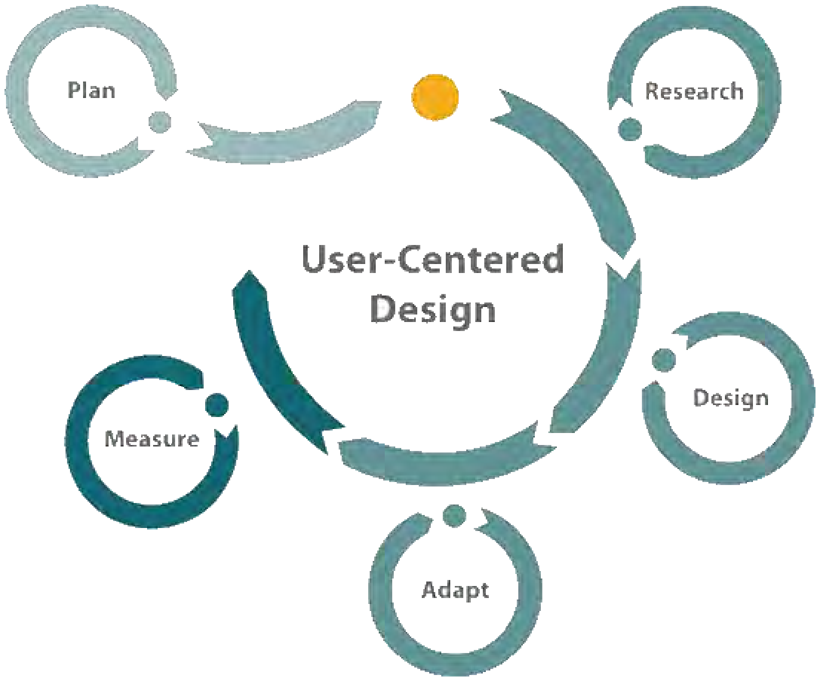


Figure 18 SAP User-Centered Design (UCD) cycle consisting of five iterative phases: Plan, Research, Design, Adapt and Measure (Weissenberger, 2009)

The development cycle follows five phases (Weissenberger, 2009):

- **Plan.** Planning is critical to the success of all projects, and this is also true of projects using UCD. In the Plan phase, the team determines all of the UCD activities and ensures that the necessary resources are available. Because of the collaborative nature of the UCD process, resources are required from multiple teams. To maximize the chances of success, it is important to plan the UCD activities for the entire project up front and to ensure that the people and budget are available for all activities.
- **Research.** Before a product can be designed, it is imperative to have a clear understanding of the users‟ goals and tasks, the market needs, and related work. Research is the second step of the UCD process. By understanding the needs of users, it is possible to uncover specific requirements for new versions, ideas for new products, and inspiration for innovation. Research projects provide opportunities for multi-disciplinary teams to gain an understanding of their users and to form a solid foundation for the Design step.
- **Design.** In the Design phase, the system is defined from the users‟ perspective. Initially, this phase takes the form of use cases and an object action model, which describes the tasks that the system will support. From these tasks UI designs can be created, beginning with rough sketches and ending with detailed UI design specifications. Taking everything learned from the Research phase - user profiles, task flows and pain points, market analysis, and the understanding of the competition - it is possible to structure the design around the way the user thinks about the system.
- **Adapt.** The UCD process does not end with the hand off of the design to the development group. The Adapt phase acknowledges that even the best conceived designs often need to be adapted when development begins coding. This adaptation can occur as a result of unforeseen limitations in the target technology, new requirements, or missing functionality in the initial design.
- **Measure.** When the product is released, it is possible to measure its usability quantitatively. These tests measure a product‟s effectiveness, efficiency, and satisfaction. Test scores are combined to provide a single number, which is the Usability Key Performance Indicator (KPI).

SAP works in a small team keeping the design and development cycles short. Prioritizing the requirements and defining deadlines for the incremental development is also included. It is very important to involve the users as much as possible (Beynon-Davies, 2000). By having conducted user interviews, observations user feedback a large user involvement is ensured. Mainly frequent releases allow constant user feedback.

For the design of the User Interface the SAP team will follow the approach of wireframing and user testing. For wireframing Balsamiq[7] will be used in order to get a quick user feedback. In parallel these mock-ups will be provided to the developers, who can start to implement the designs. With the Balsamiq tool with first ideas and user interfaces can be rapidly and efficiently designed without going too much into detail in respect to details such as precise colours, sizes and shapes.

The mock-ups look like paper prototypes, which makes it easier to get true user feedback. Often users are intimidated when polished and finished UIs are presented to them: if a team of designers and developers has been working for many months on a UI it is difficult for a User to provide an honest feedback which might put the whole effort in jeopardy. For the very first user feedback such paper prototypes are enough and don´t need to be polished.

After having defined the user interface (and gained user feedback) the UI designer works closely together with the developers: after having finished the next screens or part of the UIs, the developers sit together with the UI designer and discussed the interfaces. Consistently the results were discussed with the users. In this way the developers, designers and users collaboratively develop the first prototype.

In rapid prototyping user feedback is one of the most powerful methods. It´s very important to show the first ideas and drafts to the user as soon as possible and gather feedback, on which further designs and concepts are based. Following this approach, the designers and developers save time and money. It seems valuable to have as much iteration as possible in early design stages of a project (Landay 1995).

[7] http://balsamiq.com/

5 Conclusions

This document reports the current status of the development of the SERENOA application prototypes with a focus on requirements and design. 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
- the status of current demonstrators
- the first design mock-ups
- the process of development

Most of the topics are very specific to the tree lines of works with requirements and evaluation criteria as the exception. This reflects the effort of the SERENOA consortium to define a common ground wherever this is possible. General requirements and evaluation criteria for adaptive UI prototypes are a good candidate for such an effort. Though, many requirements have been kept general, some are again very specific for a certain line of work. Good examples are the requirements related with the business goals, which are very specific for each partner. The spaces in which the three scenarios are placed are quite diverse. The scenario by the partner TID is situated at Home, while the scenario of SAP is situated at Work with a focus on a non-office environment. The scenario of W4 happens mainly in an internet-like space.

The requirements represent different areas of expertise:

- the developers, who will finally build the application
- the end-users, who are represented by the customers of the industrial partners and
- the business analysts, who observe the market

Very closely related are the evaluation criteria, defined by experts within the consortium.

From this pool, the desired features for the application prototype need to be collected. Assuming that a priority can be given for each feature a stack of features can be built.

5.1 Future Work

The future works will concentrate on the development and the description of the prototypes. After some cycles the requirements presented in chapter 3, that have been selected from the Product Owners as features for the allocation prototypes can be assessed. The development processes will allow gathering some first user & system developer impressions on the prototypes. The ever evolving market might also reveal some similar systems/prototypes which can be compared to our SERENOA application prototypes.

The task T5.2 „Prototype Development" will also contribute and benefit from the works in task T5.1 „Generic Integrations" and task T5.3 „Evaluation".

Telefónica is set to continue the evolution of the HealthDrive application described in this document as well as the chronic patient application which will use a similar infrastructure for a slightly different process such as keeping track of the evolution of patients affected by long-term diseases at their homes, minimizing their visits to health centers. This two-way approach, the details of which will be explained in future deliverables, will require the avatar platform that we have described to be even more adaptable, thus widening the impact of SERENOA technologies in the prototypes, as well as widening the scope of SERENOA itself. All work will be backed by user and medical professional feedback to ensure the appropriateness of the design choices and eventually will be used for the Andalusian health system pilot program mentioned in this deliverable.

References

Ali, S., Lewandowski, A., Rett, J., A SOA based Context-Aware Order Picking System for Warehouses using Laser Range Finder and Wearable Computer, IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011

Aurum, A., Wohlin, C., Engineering and Managing Software Requirements, Springer, 2003

Bartholdi J., Hackmann, S., Warehouse and distribution science release 0.89, Georgia Institute of Technology, Tech. Rep., January 2009, Available Online: http://www.warehouse-science.com

Beynon-Davies, P., Design Breakdowns, Scenarios and Rapid Application Development, ECIS, 2000

Cabral et al., Approaches to Semantic Web Services: An Overview and Comparison; LNCS 3053, pp. 225-239, 2004

Cassell, J. S. (2000). Embodied Conversational Agents. MIT Press.

Cassell, J., & Bickmore, T. (2000). External Manifestations of Trustworthiness in the Interface.

Cavazza, M. S. (2010). Persuasive Dialogue based on a Narrative Theory: an ECA Implementation. International Conference on Persuasive Technology (Persuasive 2010).

Coyle, J., Bardi, E., Langley, C., The Management of Business Logistics: A Supply Chain Perspective, Cincinnati, OH: South-Western College, 2002

Landay, J. A., Myers, B. A., Interactive Sketching for the Early Stages of User Interface Design, Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press, 1995

Rosson, M. B., Carroll, J. M., Usability Engineering – Scenario-based Development of Human-Computer Interaction, 2002

Schwerdtfeger, B., Klinker, G., Supporting Order Picking with Augmented Reality, IEEE International Symposim on Mixed and Augmented Reality, Cambridge, UK, 2008.

Siewiorek, D., Smailagic, A., Bass, L., Siegel, J., Martin, R., Bennington, B., Adtranz, A mobile computing system for maintenance and collaboration, International Symposium on Wearable Computers, IEEE Computer Society Press, 1998, pp. 25–32

Siewiorek, D., Smailagic, A., Starner, T., Application Design for Wearable Computing; San Rafael, CA: Morgan Claypool, 2008

Smailagic, A., Siewiorek, D., Martin, R., Stivoric, J., Very rapid prototyping of wearable computers: a case study of custom versus off-the-shelf design methodologies; Design Automation for Embedded Systems, vol. 3, no. 1, pp. 217–230, March 1998

Starner, T., Wearable computers: No longer science fiction; IEEE Pervasive Computing, vol. 1, no. 1, pp. 86–88, 2002.

Stein, R., Ferrero, S., Hetfield, M., Quinn, A., Krichever, M., Development of a commercially successful wearable data collection system, IEEE International Symposium on Wearable Computers, IEEE Computer Society, 1998

Sundar, D., Software Engineering, Laxmi Publications, 2010

Weissenberger, U., Fellenz Thompson, C., SAP User Experience, SAP AG, 2009

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

Annex I: Selected Requirements from D1.1.1

Table 1: Non-functional (NFR) and Functional (FR) Requirements from UI Developers, Consumers and Business

Name	Priority	NFR	FR	Domain	Index
Accessibility	High	X		Consumer	5
Adaptation	High	X		Developer	19
Anticipation of Events	Medium	X		Dev.+Con.	19.8
Audio Feedback	High		X	Consumer	5.2
Environment-dependent Adaptation	High	X		Developer	19.5
Control Over the Adaptation Process	High	X		Dev.+Con.	19.1
Cross-Platform Consistency	High	X		Consumer	4
Customization	High		X	Consumer	1
Deadline-driven		X		Business	2
Dynamic User Choice	High		X	Consumer	4
Easy Connection and Configuration	High	X		Consumer	6
Efficiency	High	X		Consumer	11
Error	High	X		Consumer	2
Feedback	High		X	Consumer	5
Identification	Medium	X		Consumer	22
Independence of Different Technological Spaces	High	X		Consumer	8
Intuitiveness	High	X		Consumer	18
Learnability	High	X		Consumer	10
Memorability	High	X		Consumer	12
Multimodality	High	X		Developer	4
Multimodality	High	X		Consumer	7
Performance	High	X		Consumer	1
Personalization	High		X	Consumer	2
Platform-dependent Adaption	High	X		Dev.+Con.	19.2
Response Time	High	X		Consumer	16
Satisfaction	High	X		Consumer	13
Search	Medium		X	Consumer	6
Should Support Several Adaptation Techniques	High	X		Dev.+Con.	19.4.
Simplicity	High	X		Consumer	14
Stability	High	X		Consumer	17
System and Task Continuity	High	X		Dev.+Con.	19.6.
Time to Market		X		Business	1
User-dependent Adaption	High	X		Dev.+Con.	19.3

Name	Priority	NFR	FR	Domain	Index
Visual Feedback	High		X	Consumer	5.1
Working environment	High	X		Consumer	3