Multi-Dimensional Context-Aware

Adaptation of Service Front-Ends

Project no. FP7 – ICT – 258030

Deliverable 2.4.2 Criteria for the Evaluation of CAA of SFEs (R2)

Due date of deliverable: 31/08/2012

Actual submission to EC date: 31/08/2012

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)		
[PU]	[Pubic]	Yes

Document Information	
Lead Contractor	CNR-ISTI
Editor	Fabio Paternò, Carmen Santoro, Lucio Davide Spano
Revision	v.5 (31/07/2012)
Reviewer 1	TID
Reviewer 2	UCL
Approved by	
Project Officer	Michel Lacroix

Contributors	
Partner	**Contributors**
CNR-ISTI	F. Paternò, C. Santoro, L. D. Spano

Changes			
Version	**Date**	**Author**	**Comments**
1	17/07/2012	Carmen Santoro	TOC
2	24/08/2012	F. Paternò, C. Santoro, L. D. Spano	Version for Internal Review
3	27/08/2012	F. Javier Caminero	TID review
4	28/08/2012	Vivian Motti	UCL review
5	29/08/2012	F. Paternò, C. Santoro, L. D. Spano	Final Version

Executive Summary

The main goal of this deliverable is to define criteria (both qualitative and quantitative) and potential benchmarks for evaluation of tools for Context Aware Adaptation (CAA) of service front ends (SFE) and the associated applications. These criteria are expected to provide useful indications for those who work in the field of context-aware adaptation of SFEs and become the backbone for systematic ongoing scientific research. The criteria will be applied first internally to the languages and tools developed in the project in order to conduct self-evaluation of the progress, but also externally with respect to competitors in order to provide a comparative analysis. Evaluation protocols and potential benchmarks will be made publicly available allowing other laboratories to compare results and advance the state-of-the-art.

The criteria will allow the assessment of the proposed solutions both from the designers and the end-user viewpoints in terms of their effectiveness and satisfaction. We will consider usability of the various project results related to CAA, including the authoring tools and the adaptation at run-time. The usability of the model-based approaches will be investigated as well. The adaptation process will also be evaluated in terms of software engineering parameters (such as robustness, efficiency, portability, etc.)

Table of Contents

1 Introduction

1.1 Objectives and Overview

The main goal of this deliverable is to define criteria (both qualitative and quantitative) and potential measures for their evaluation in tools for Context-Aware Adaptation of service front ends and the associated applications.

1.2 Audience

The audience for this deliverable is twofold: *the project team*, in order to provide concepts, methods, and tools that can continuously provide feedback on how well adaptation is supported; *the broader international developers and designers community*, that can be interested in applying such evaluation criteria to their context-sensitive tools and applications as well.

1.3 Related Documents

- Deliverable D1.1.1 Requirements Analysis provides some useful input for identifying the evaluation criteria.
- Deliverable D1.1.2 Requirements Analysis (R2) provides some useful input for identifying the evaluation criteria.
- Deliverable 2.1.2 CADS and CARF provides a set of qualities of software that are relevant while executing adaptation
- Deliverable D5.2.2 is a first version of a deliverable about the application prototypes.
- Deliverable D1.2.1, on the architecture, provides useful indications about the project results that will have to be assessed.
- A deliverable on the same topic was provided for M12 (D2.4.1), which indicated a first set of criteria that can be relevant for the project. This deliverables aims to provide a self-contained update of such set of criteria.

1.4 Organization of this document

Section 1 (this introduction) presents the goals, audience, related documentation of this deliverable and the organisation. In Section 2 we provide an overview of what will have to be evaluated in the Serenoa project (design and run-time tools, languages for user interfaces, applications…). We then have a dedicated section focused to a review of the requirement specification, in order to synthesize some useful evaluation criteria (Section 3). Section 4 is dedicated to the technical evaluation. Section 5 is about the usability evaluation and Section 6 revises the criteria more specific to User Interface adaptation.

2 What will be Evaluated

In Serenoa, we plan to evaluate the adopted solutions from two points of view: a technical evaluation using some relevant software quality factors , and also a usability and accessibility evaluation by considering criteria aimed at evaluating user-oriented aspects.

We have identified a number of categories for the project results that will be subject to evaluation:

- The various software tools that will be developed within the project (e.g. tools for developing UIs, tools for specifying adaptation rules).
- Applications (the applications that will be generated using the above tools).
- Requirements (the requirements that have been identified by the project will be checked in order to evaluate to what extent they have been satisfied).
- Models and languages (UI models and related specification languages produced by the project will be evaluated as well).
- Architectural modules/components (specific architectural modules that will be subject to evaluation).
- Other possible solutions and algorithms (solutions/algorithms developed by the project and which can be evaluated "per-se", even without the existence of an underlying supporting tool).

In general, these results will be subject to both a technical evaluation and a usability evaluation. When we deal with a user-oriented evaluation, we have to assume that two main types of users will be considered: interactive application developers and end-users, depending on the type of system considered. The tools that will be considered are:

- The authoring tools for developing multi-device interactive applications at design time (for which we plan to carry out both a technical and a usability evaluation for developers),
- Tools for specifying adaptation rules at design time (for which we plan to carry out both a technical and a usability evaluation for developers).
- Tools for performing adaptation of interactive applications. In this category we can find both tools that adapt user interfaces within a given modality (e.g. from graphical desktop to graphical mobile devices) and tools that adapt to a different modality (e.g. from graphical Web user interface to vocal interface),
- Tools for customizing adaptation rules (which can be applied either at design time or run time by developers, but in some cases even by end users).

Regarding the applications, we can distinguish:

- Adapted applications in which the various adapted versions are produced by authoring tools at design time.
- Adapted applications in which the adapted versions are produced by runtime adapters.

In both cases the attention will be on an end-user evaluation in terms of usability and accessibility. In the Serenoa deliverable *D1.1.1-Requirements Analysis (R1)*, a number of functional and non-functional requirements for developers and end-users have been put forward. Such requirements will be considered in the evaluation as well. They provide some useful input for the evaluation criteria and can be further refined in the evaluation work, and in the end we will be able to indicate which of them are fulfilled.

Another important output of the project will be a set of model-based languages for multi-device interfaces and adaptation. This is also an input for standardization in the W3C. Some partners provide useful input for this work, such as MARIA at CNR-ISTI, UsiXML at UCL, MyMobileWeb at CTIC and LEONARDI at W4. Such languages can be assessed in technical terms and also in terms of usability for developers. For example in (Souchon and Vanderdonckt, 2003), five criteria for assessment have been identified: abstraction level, amount of tags, expressivity of the language, openness of the language, coverage of concepts. Other software components will be developed in the project, such as the context manager. Since this will be an infrastructure for supporting the overall adaptation environment its evaluation will be mainly technical, e.g. in terms of its potential ability to support various types of sensors and devices. Lastly, the algorithms for adaptation can be considered in the evaluation work as well. In this case the goal will be to compare different solutions for the same adaptation issues in terms of performance and coverage of the relevant cases.

3 Requirements and Evaluation Criteria

In the previous version of this deliverable (namely, *D2.4.1-Criteria for evaluation of CAA of SFEs (R1)*) we already introduced some preliminary considerations of evaluation criteria with regard to projects' identified requirements by highlighting some relationships between the evaluation criteria and the requirements/scenarios identified in D1.1.1.

In this section we revise the list of requirements contained in the updated version of project requirements (D1.1.2) and take it as a source of information for deriving further evaluation criteria for CAA of SFE. Therefore, in this section we discuss some requirements specifically related to adaptation and from which we can derive further evaluation criteria. As you will see, some requirements have been already associated with evaluation criteria already identified in this document.

3.1 Requirements and validation criteria for various Serenoa results/architectural components

3.1.1 Authoring Tools

In D1.1.2 we identified a number of requirements concerning the authoring tools.

One requirement states that *the authoring tools developed in Serenoa must support the creation and editing of adaptation rules at any abstraction level (abstract, concrete & implementation).* In this case the requirement is focused on the specification and authoring of the adaptation rules. In addition, this requirement is also connected with the exploitation of different UI abstraction levels, which we specify through model-based approaches.

Therefore a new evaluation criterion can be identified in the a*bility of specifying adaptation that affects the UI at different levels of abstraction.*

With this criterion we want to check that multiple abstraction levels can be handled by adaptation, enabling the designer to define the effects at different levels of abstraction. A possible way for evaluating the criterion is counting the number of levels of abstraction that can be addressed using the authoring tools. As acceptance criterion, we should ensure that it is possible to define adaptations affecting the various UI levels. Consequently as acceptance criterion, the authoring tools have to make it possible to define adaptation rules affecting all the various UIs.

Another requirement stated that *the authoring tools developed in Serenoa must support the authoring of multi-device interactive applications*.

The authoring tools developed in Serenoa should facilitate designers/developers in specifying/creating interactive applications targeting different platforms and exploiting different modalities. Therefore a new evaluation criterion can be identified in *Multi-platform adaptive application design*: the authoring tools developed in Serenoa facilitate the authoring of applications which are adapted to be run on various platforms. A possible measure for this criterion is the number of platforms that the adaptation performed by the Serenoa Framework is able to support. As acceptance criterion, at least desktop, mobile, vocal and multimodal (combination of graphical and vocal) modalities should be addressed.

3.1.2 Context Manager

In D1.1.2, there is a number of requirements related to handling context information and more specifically to the features that the Context Manager (the Serenoa architectural component devoted to capturing, handling, updating and notifying/sending context information to other architectural modules, saving context models, etc.) should provide. These requirements can be related to the criteria we describe in Section 6.3.1 of this document and related to context (e.g. context representation, modelling, etc.) In particular, the adaptation supported in Serenoa should target to manage context information considering various aspects: user, environment, technology and social aspects. Therefore, this requirement is connected with the evaluation criterion called "Coverage/Expressiveness of the context model" described in Section 6 of this document. Other requirements are covered by more general Technical Evaluation Criteria such as *Interoperability* (see section 4.1) and *Correctness* (see section 4.2).

3.1.3 Advanced Adaptation Logic Description Language

In D1.1.2 we also have a number of requirements related to the specification of the AAL-DL language which is aimed to describe adaptation rules. We have to apply the same criterion described in the first part of section 3.1.1 to the AAL-DL language. Indeed, if the authoring tool should provide means for the application designer to define adaptations that affect different levels of abstractions, the AAL-DL language should be able to capture such definition in a format which is independent from the authoring tool.

Various requirements described in D1.1.2 are related to the different types of Transformers and Generators that we consider in the Serenoa architecture: Vocal/Mobile Web/Avatar Transformer (e.g. the Vocal Transformer takes a CUI for a desktop platform and transforms it into a CUI for the vocal platform) and Vocal/Mobile Web/Avatar Generator (e.g. the Vocal Generator generates a Final UI, starting from a concrete UI description for the vocal platform). A requirement that was stated for these modules was the possibility for users to customise the transformations carried out by these modules. This criterion can be connected to the property of controllability/programmability of the adaptation which we will better describe in Section 6 of this document, and grouped under the user-oriented criteria (as it is a criterion that targets the end-user perspective).

In D1.1.2 there is also a requirement concerning the performance of the generation process, which has been stated for e.g. the Mobile Web Generator as: “The performance of the generation process should be sufficient to guarantee the simultaneous access of several clients with an acceptable response time.” This requirement can be exploited to derive a new evaluation criterion for the adaptation, namely the fact that the adaptation should be able to cope with scalability issues. Indeed, when multiple users/clients exploit the Serenoa solutions for adaptive UIs, the performance of the system should not be compromised. So, this requirement can be exploited to derive a new evaluation criterion: *Scalability of the solution providing adaptivity,* which we included under the Technical Evaluation Criteria in section 4.13.

3.1.4 Adaptation Engine

The purpose of the Adaptation Engine is selecting from the description of SFEs the suitable advance adaptation logic required to adapt to the active context. One of the main functional requirements of this module as stated in D1.1.2 is to “empowering the Runtime Engine through the most suitable adaptation logic in function of the active context.” Indeed, this module should be able to select the optimal AAL rules/actions based on the user context. This requirement can be seen as connected with the adaptation property called “Appropriateness of the adaptation”, described in section 5 of this document.

Another important aspect to be evaluated for the Adaptation Engine is the ability of understanding the definition of the adaptation logic, described with the AAL-DL syntax. Therefore, it has to be assessed not only the possibility of defining such logic (as we already discussed in section 3.1.1), but also the ability of the application developed using the Serenoa Framework to adapt accordingly. Therefore, we established as a validation criterion the a*bility of adaptation to affect UI at different levels of abstraction.* This means that Adaptation Engine should be able to select actions defined for higher level of abstraction (e.g. the Abstract User Interface) and whole adaptation system should be able to interpret and perform them. It is possible to measure this criterion counting the number of levels of abstraction that can be managed while performing the adaptation. As an acceptance criterion, it should be possible to interpret the adaptation logic defined at all the levels of abstraction.

In addition, this module, as stated in D1.1.2 should be able to change its execution parameters based on feedback . This requirement can be exploited for deriving an additional criterion for adaptation, namely the possibility that adaptation can *learn* from the user behaviour.

3.2 Requirements and Validation Criteria for the Serenoa Prototypes

For each prototype considered in Serenoa, a new version of requirements has been delivered in D5.2.2. For each requirement an associated validation/acceptance criterion has also been identified in that deliverable. It is also worth pointing out that some requirements are shared among the various prototypes, therefore, we will consider them only once in this document.

3.2.1 E-Health scenario

Requirement 1.3 (**Multimodality**) refers to the possibility, in the e-Health scenario, to support a "graceful combination of several modalities". For example, in this particular scenario it would mean the coordination of visual data (e.g. medical graphs) with the avatar behaviour (e.g. gestures or voice). From this requirement we can derive an additional point of evaluation for the *UI Languages Expressiveness* criterion, which states that the languages must support the definition of multimodal interactive applications, described in Section 4 of this document.

Requirement 1.5 (**Control over the adaptation process**) refers to the fact that the user has the possibility to customise the resulting avatar: for instance, the user can select to have a fully-fledged avatar and/or use just the speech version without a physical visualization of the character. As a method for evaluation we use inspection: the requirement is planned to be evaluated by checking if the user always has the possibility to select one of the aforementioned possibilities (avatar-based and/or voice-based). This requirements is highly connected with the evaluation criterion named "*Controllability/Customisability of the adaptive behaviour*" described in Section 6 of this document.

Requirement 1.6 (**Cross-Platform Consistency**) refers to the possibility that, although it will not be possible to keep the avatar characteristics in devices across the board, extra effort will be put in delivering a consistent avatar, even if the functionality is compromised in some cases. As an example, videos of the avatar may be used, but even in those videos the look and feel and behaviour of the avatar should be kept consistent. As stated in D5.2.2, a questionnaire will be used for evaluation. Users' perception about the avatar's identity (i.e. look-and-feel, behaviour, etc.) running on different devices will be requested. This requirement is closely connected with the criterion named "Across-device consistency" and described in Section 6 of this document.

Requirement 1.9 (**User-dependent Adaptation**) refers to the fact that the system, including the avatar, will adapt automatically to the needs of different users. An inspection-based method should be used for evaluation (see D5.2.2): it will be seen as fulfilled when the system adapts to at least one user-characteristic (e.g. language, experience, preferences, etc.). Therefore, with these requirements we want to ensure that the adaptation solution caters for the different needs of their users to guarantee a good user experience. This requirement is connected with the "*Controllability/Customisability of the adaptive behaviour*" described in Section 6 of this document.

3.2.2 E-Commerce transaction scenario

Requirement 2.1 (**Easy Connection and Configuration**) refers to the fact that the system will enable the end users to configure their environment easily. Inspection: all available platforms need a fast and easy way to login to the system and access the settings menu. This requirement is connected with the *Comprehensibility* and *Controllability/Programmability* criteria described in Section 4 from a technical point of view. While from the usability side, it is connected to the *Comprehensibility* and_*User's Awareness of UI Adaptation*, described in Section 6.

Requirement 2.3 (**Platform-dependent Adaption**) states the fact that the application will run on mobile devices. In addition, it states that the application will run at least in one of the following mobile operating systems: iOS and Android. This requirement is connected with the *Portability* criterion, described in Section 4.

Requirement 2.6 (**Accessibility**) states that the system will be accessible for at least one specific group of users with at least one of the following constraints: i) users speaking different languages than the default language; ii) motor disabilities; iii) cognitive disabilities; iv) elderly people.

By analysing this requirement we might want to add a new evaluation criterion checking to what extent the adaptation is able to provide accessible UIs: *Accessibility of the adaptation,* described in Section 4.

3.2.3 Warehouse Management scenario

Requirement 3.11 (**Working Environment**) states that the system will be adaptive to the working environment. The requirement will be seen as fulfilled when the system adapts to at least one aspect of the environment (e.g. light or sound conditions, location of the worker, etc.) With this requirement we want to evaluate whether the application is appropriate for use in various contexts. This requirement is connected with the *Effectiveness* and *Flexibility* criteria described in Section 6.

Requirement 3.13 (**Cross Platform Consistency**) states that the application will run consistently across different platforms. The requirement is fulfilled if, through an inspection, no major differences can be found between the interfaces, interactions styles and the functionalities offered across the different platforms. This requirement is connected with the *Consistency of the Adaptation,* in particular to the *Across*-*Device Consistency* criterion, discussed in Section 6.

4 Revised Technical Evaluation Criteria

With technical evaluation, we plan to assess criteria associated with the features that are supported by the tools, tool performance, as well as the technological solutions and algorithms that have been identified in the project. In the scientific literature, a number of technical metrics coming from the software engineering field (see (ISO/IEC 9126) and (van Vliet, 1999) have been reported. In general, various aspects can be considered for each criterion:

- *Applied to*: for each criterion it can be useful to indicate to what it is expected to be applied. Possible values for this attribute could be: all the tools (if the criterion can be applied to evaluate all the tools developed in the project), a specific tool type (if the considered criterion is relevant only for some specific tool type), specific architectural components, UI languages, etc.
- *Criterion name*: it is important to provide a meaningful name for the evaluation criterion.
- *Definition*: a definition of the criterion, together with further explanations, if needed, for clarifying the associated concepts.
- *Measure*: a measure which can provide a concrete, verifiable variable to be associated with the criterion. For instance, a way of quantifying interoperability could be the number of interoperable data formats that a tool can support.
- *Technique/Method of application:* possible technique(s) with which data associated with a certain criterion can be gathered.
- *Success/Acceptance*: it can be useful to determine whether a certain criterion goal has been achieved or not. It can be seen as a sort of threshold for deciding about whether the criterion has been met or not.

In the following sections we introduce a number of criteria that are relevant for our project and discuss them taking into account such aspects, when possible.

4.1 Interoperability

Interoperability is the capability of a set of tools to work together. It provides a measure of the ability to exploit some information across the different tools.

Interoperability can be measured by data exchangeability, i.e.: the number of interoperable data formats used in the tool. Another possible measure is the number of platforms/tools (both developed in the project and relevant third-party tools) that can interoperate. In the project, we plan to identify and count the platforms in which the Serenoa results can work. As acceptance criterion, the tools developed inside the Serenoa project must be able to exchange the information about the adaptation logic, the description of the interactive applications and the information about the context of use. A certain level of interoperability must also be addressed across the languages and their underlying models. For instance, the AAL-DL should only provide means for defining the adaptation rules, while the concepts related to the context or the user interfaces are taken by other languages (e.g. the context and the ASFE-DL).

4.2 Correctness

Correctness is the degree with which software performs its tasks as defined in the requirements specification.

For assessing this criterion, it is useful to calculate the *Must-have* and *Should-have* requirements that have been respectively *partially* or *totally* fulfilled. It can be considered successfully verified if 100% of Must-have requirements have totally been fulfilled and if more than 50% of Should-have requirements have partially OR totally been fulfilled. The criterion applies to all the project outputs that have associated requirements (see D1.1.2).

4.3 Adherence to standards

Adherence to standards is the ability of software to comply with standards, regulations, guidelines, conventions, etc. Thus, the measure of it requires counting the number of relevant standards, guidelines, etc., the system is compliant with. In Serenoa, we plan to consider in particular the W3C standards, such as the guidelines for accessibility, some usability standards, and also to create new standards for task model and abstract user interface model descriptions.

4.4 Portability

Portability is related to the capability of the software product to be transferred from one environment to another.

Thus, it requires an analysis of the number of different (HW/SW) environments the system supports, whether the software depends upon libraries unique to a particular HW/SW installation, how much effort would be required to transfer the program from one HW/SW environment to another. For example, in Serenoa the authoring environments should be able to run on Windows, Linux and Apple OS X and the adaptation to mobile devices should support at least the main mobile platforms (i.e. iOS and Android). As acceptance criterion, the software developed inside the Serenoa project should be portable with a reasonable effort for all those platforms that have been selected for the applications scenarios.

4.5 Efficiency

The capability of the software product 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.

This means the degree with which software fulfils its goal without waste of resources/time (e.g. whether functions have been optimized for speed). We can distinguish: *Time behaviour*, capability of software to provide appropriate response and processing times when performing its functions under stated conditions and *Resource utilisation*, e.g. usage of memory for saving data. Acceptance criterion: usage of time and resources that do not degrade the user experience.

4.6 Maintainability/ Changeability

Maintainability and changeability refer to the capability of the software product to be modified after a working version is delivered. Modifications may include corrections, improvements or software adaptation to changes in environment, in requirements and functional specifications. The modification to one module should have as less impact as possible on the others. It should be possible to modify the code without introducing unexpected errors.

This refers how easily changes can be made to satisfy new requirements or to correct deficiencies. Possible measures are:

- Mean Time To Change (MTTC), when an error is found, it measures how much time it takes to analyse the change, design the modification, implement and test it;
- Cost of change/total cost of system;
- Readability of source code (see Aggarwall et al., 2002);
- Documentation quality (see Aggarwall et al., 2002).

In our project it refers to the easiness in which it is possible to modify the languages for interactive applications or adaptation rules, or the instances of such languages.

4.7 Extensibility/ Evolvability

Extensibility and evolvability refer to the ability to extend a system at minimum effort cost. For example, in an adaptive system it can be the ability to add new adaptation rules or modify old ones.

4.8 Modularity

Modularity refers to the logical partitioning of the software design that allows complex software to be manageable for the purpose of implementation and maintenance and that enable parallel work.

The degree of independence of the various components can be measured by Cohesion, i.e. by evaluating how well components of a module fit together or, in other words, the degree to which all elements directed towards the same task are contained in a single component. The Cohesion should be maximized.

Another possible measure is Coupling, which indicates how much different modules have to communicate. In this case, Coupling should be minimized.

4.9 Reliability

Reliability refers to the capability of the software product to maintain a specified level of performance when used under specified conditions.

In this case, the measures can be: Mean Time Between Failure (MTBF), the frequency of software failure is measured by the average time between failures; Mean Time To Repair (MTTR), the criticality of software failure is measured by the average time required for repair (for example the time to restart an adaptation server) and Reliability ratio (MTBF/ MTTR).

4.10 Availability

Availability refers to how long a system is operational. This can be measured by the percentage of time a system is available, versus the time the system is needed to be available. Formally, the system availability can be defined as MTBF/(MTBF + MTTR).

4.11 Scalability

Scalability refers to 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. For example, the model-based languages that we are going to adopt in Serenoa should be able to allow designers to specify even complex applications and not only small examples.

4.12 Testability

Testability refers to the capability of the software product to allow modified software to be validated. It can be supported through a set of built-in test functions.

4.13 Recoverability

Recoverability refers to the capability of the software product to re-establish a specified level of performance and recover the affected data in the case of a failure. It can be measured with the MTTR (Mean Time To Repair).

4.14 Languages Expressiveness

The expressive power of the language refers to the degree to which the languages can describe every aspect relevant to the domain considered. In our project we consider languages for interactive applications (which should be able to describe presentation, dynamic behaviour, content), and languages for the adaptation logic that should be able to indicate when adaptation should be triggered and what effects will generate.

5 Revised Usability Evaluation Criteria

In this section, we provide a number of criteria aimed at evaluating user-oriented aspects. It is worth noting that all such criteria (both technical and user-oriented metrics) are very often dependent on each other. Then, it may happen that when increasing the compliance with one criterion, the compliance of other criteria can be increased as well. For instance, for technical criteria, when increasing modularity, maintainability will also likely be improved. Similarly, comprehensibility is strongly related to developer learnability. However, in other cases, this is not true: for instance, sometimes, in order to increase security, the resulting usability of the system decreases, or end user learnability is not directly impacted by software modularity. Therefore, taking into account such relationships will be important to reach a reasonable trade-off for such criteria, also taking into account the most relevant criteria for the project goals.

The ISO standard definition of usability is *"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use"* (ISO/IEC 9241-11). In addition other aspects can have an impact on the overall usability of a system.

Usability criteria affect both the authoring tools and the applications created with the Serenoa Framework.

5.1 Effectiveness

Effectiveness refers to the accuracy and completeness with which certain users can achieve specified goals in particular environments.

This can be measured in terms of task success rate: for each task, the average percentage of users who successfully completed each task or in terms of percentage of users who were able to complete all the tasks. The actual task completion can be detected in various ways: by an external observer, by explicit indication of the user or through the automatic detection of the events associated with it. For every unsuccessful task, in order to increase the test reliability, the cause of the failure should be considered too. Errors caused by the user (e.g., wrong interaction) should be distinguished from application errors (bugs) and from the abandonment of the task (that takes place when the user refuses for some reason to complete it).

5.2 Efficiency

Efficiency refers to the resources consumed in relation to the accuracy and completeness of goals achieved.

The system should be efficient to use, which means that once the user has got familiar with it, a high level of productivity is possible. One possible manner to measure it is through the task completion time (or time-on-task): time (in seconds) between the start and the end of a task, measured for each task and user. This can be measured in various ways: a) Automatic data logging during user tests; b) Moderator/note taker uses a stopwatch to measure such times; c) Evaluator reviews a video recorded during the user testing. Another measure is the average number of errors made by each participant for each task. An error is an action useless for performing the current task. Also this measure can be detected by: a) Automatic data logging during user tests; b) Moderator/note taker observes the user during the test; c) Evaluator reviews a video recorded during the user testing. Other possible measures are: the time spent on errors and the frequency of help use (i.e. the number of times users used the help of the system, measured for each task and average).

5.3 Satisfaction

Satisfaction refers to the comfort and acceptability offered by the work system to its users and other people affected by its use.

This can be measured by the number of times the user expresses clear frustration or joy, which can be detected in several ways, e.g.: a) Moderator annotates the information while observing the user during the test; b) Evaluator reviews the recorded video. Other measures can be given by after-test questionnaire (filled in immediately after task performance and at the end of a usability test session); Rating scale for satisfaction with functions and features, which can be done through: a) Satisfaction surveys or qualitative interviews to get user attitudes towards the software; b) System Usability Scale (SUS, a ten-item attitude Likert scale giving a global view of subjective assessments of usability). Another relevant measure is the percentage of users that rate the product as better than a key competitor.

5.4 Learnability

Learnability refers to the capability of the software product to enable the user to easily learn how to use it.

It measures how much time and effort are required to become proficient with the software. It refers to the novices' ability to reach a reasonable level of performance rapidly.

It is possible to have Subjective measure (user feedback) in terms of a (often 5-point) Likert rating scale, which can be provided by an after-test questionnaire response. Another Subjective measure is the number of learnability-related user comments, which can be detected by: a) Moderator annotates the information while observing the user during the test or b) Evaluator reviews the recorded video. There are also metrics that consider effectiveness of documentation and help, such as: decrease in help commands used over certain time interval; what proportion of tasks can be executed correctly after using documentation/help; help content easy to navigate. Other metrics are based on usability change, which requires assessing and comparing the change in usability over time (e.g. improvements in efficiency in different trials over time). A further measure is "How long does the user take to learn how to perform the specified task efficiently?" (i.e. time to achieve expert performance) or to compare the usability of a product for novice and expert users.

5.5 Memorability

Memorability refers to 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. This requires measuring retention over time, which means that it should not take more than a certain amount of time depending on the system used to re-learn. Thus, it is required to test the system with the same users in different trials over time.

5.6 Comprehensibility

Comprehensibility refers to the capability of the software product to enable the user to understand a) whether the software is suitable, and b) how it can be used for particular tasks and conditions of use.

This can be measured through a Likert rating scale, which can be included in after-test questionnaires.

5.7 Error tolerance

Error tolerance refers to the ability of the software to provide users with relevant support in case of users' errors, in order to allow an easy recovery. An error is an action performed by the user that is not useful to the accomplishment of the current task.

Error tolerance can also be measured through a Likert rating scale, which can be included in after-test questionnaires.

5.8 Accessibility

Accessibility refers to the degree to which a product, device, service, or environment allows as many people as possible, including persons with some level of impairment, to access the relevant information.

This can be measured through checking the capability of the software product to adhere to standards, conventions, style guides or regulations relating to accessibility (such as W3C Accessibility standards).

5.9 Attractiveness

Attractiveness refers to the capability of the software product to be attractive to the user. This criterion can be used to measure whether or not the UI uses appropriate graphical design solutions. It is important, as it is a factor which affects the user's perception of the system. It can be assessed by asking users to rate it based on a given scale (e.g. Likert rating scale), which can be included in after-test questionnaires. The measures used are subjective.

5.10 Customization

Customization refers to the degree of control the software provides to the user.

It deals with how users feel that they are in control of the software. It can be measured through a Likert rating scale provided by after-test questionnaire or by measuring the percentage of aspects that can be controlled. For example, it can be possible to measure to what extent the adaptation rules can be customized.

5.11 Flexibility

Flexibility refers to the multiplicity of ways the user and the system exchange information (Input/Output) in different forms without fixed task ordering.

5.12 Effective feedback

The system should offer informative and effective feedback about the effect of the interaction. The effectiveness of the feedback can be measured by submitting questionnaires to users asking about the provided feedback. Otherwise, different feedback types can be compared by users in order to assess the most preferred one.

5.13 Tool support for usability evaluation

Usability evaluation is an important phase. For this purpose automatic tools are very useful to gather larger amount of usability data and support their analysis. In particular, it is possible to enable remote evaluation (Hartson et al., 1996), which implies that users and evaluators are separated in time and/or space, including some functionalities that gather the needed data in the developed tools. This is important in order to analyse users in their daily environments and decreases the costs of the evaluation without requiring the use of specific laboratories and asking the users to move. In particular, tools for remote Web usability evaluation should be sufficiently general so that they can be used to analyse user behaviour even when using various browsers or applications developed using different toolkits.

Ivory and Hearst (2001) provided a good discussion of tools for usability evaluation according to a taxonomy based on four dimensions: method class (the type of evaluation); method type (how the evaluation is conducted); automation type (the evaluation aspect that is automated); and effort level (the type of effort required to execute the method). CNR-ISTI is developing a tool, Web Usability Probe (Carta, Paternò, Santana, 2011), which allows the evaluator to create tests based on real Web sites instead of prototypes, giving the possibility to observe the user's interaction in a real environment. According to Ivory and Hearst classification, this solution is for usability testing, it captures logs generated client-side, supports automatic analysis and a number of visualizations to ease the identification of the usability issues, and only requires that users perform some predefined tasks specified by the evaluators. The tool works also for mobile applications, thus it can be useful to support usability evaluation of Web applications accessed through various types of interactive devices.

5.14 Actual system adoption

This criterion should evaluate the extent to which a particular system is actually used. It can be measured, in a long period, by counting the number of times users have voluntarily used the system.

5.15 Intention to use the system

This criterion should evaluate the intention of a person to use a particular system. This criterion seems to have relationship with the perceived usefulness of the system, the easiness in using the system itself and also the likeability of the system (to what extent the user liked/was satisfied by using the system) that the user might have perceived the first time s/he interacted with the system. By knowing how the user assessed these criteria could allow to more easily predict the intention the user has to use the system. Differently from the previous criterion, this criterion evaluates the intention of the user to use the system in the long term.

6 Revised UI Adaptation-Oriented Criteria

Adaptation can be applied to various user interface aspects:

- *Presentation*, for the perceivable aspects (including media and interaction techniques choice, layout, graphical attributes, …)
- *Dynamic behaviour*, including navigation structure, dynamic activation and deactivation of interaction techniques, …
- *Content*, including texts, labels, images.

Due to the importance of UI adaptation in our project, in this section we focus on criteria specifically relate to this aspect.

In the following sections, for each identified criterion, we provide its definition, a suitable measure for assessing it as well as suitable acceptance criterion, and techniques/methods that can be used for collecting the related data. Indeed, when considering evaluation issues, a number of questions should be answered. First of all, what has to be evaluated (e.g. tools, applications, models ...), then the evaluation criteria and the associated measures (how to measure the criterion), the success/acceptance criterion (what are measures of success?), evaluations methods (how to collect evaluation data?). Of course different evaluation methods can differ according to a number of aspects, for instance when the evaluation is done, how the evaluation is done, by whom the evaluation is done. In addition, evaluation criteria can include both qualitative (user's perceived feedback on e.g. application features) and quantitative measures (e.g. task completion time, system's latency time...). There are a number of related work connected with adaptive systems and how to evaluate them (Gena, 2005; Paramythis, 2010). In the following section we summarise the main criteria we identified by also reviewing the state of the art.

In particular, in the following we divide the criteria related to adaptation in a number of groups. It is worth pointing out that different grouping criteria can be also used. For instance Paramythis et al (2010), as well as Brusilovski et al. (2004) considered a layered evaluation of adaptive systems and then in this case the evaluation criteria are grouped in terms of the adaptation phases identified by the authors. In (Van Velsen et al., 2008), authors divide the criteria they identified in 4 categories: (1) variables concerning attitude and experience (e.g. trust and privacy issues, user experience...); (2) variables concerning actual use (e.g. usability, user performance...) (3) variables concerning system adoption (e.g. intention to use, perceived usefulness...) (4) variables concerning system output (e.g. appropriateness of adaptation, comprehensibility or unobtrusiveness).

We mainly identified the following groups:

- *Technical* criteria: criteria that can be objectively measured by just considering the support offered by the system.
- *User-oriented* criteria: criteria that refer to perceived user feedback on a number of aspects connected with adaptation.
- *Modelling-related* criteria: in this group we put the criteria related to adaptation which are specifically related to the activity of modelling e.g. the context (and whose results are used by the adaptation process).

6.1 Technical Criteria

6.1.1 Adaptation Granularity

This refers to the breadth of granularity levels that adaptation can support. Indeed, adaptation (in terms of e.g. adaptation rules) can have mainly three granularities, from having an impact on the whole UI, to the finest grain of UI elements. More in detail, we can have either a complete change of the UI (e.g. from vocal to graphical if some contextual conditions change, e.g. the environment becomes noisy), or a change of some user interface parts, or even a change of attributes of specific user interface elements (e.g. change of font size of a textual element). This criterion is one of the CADS (Context Aware Design Space) dimension for the adaptation design that is described in D2.1.2 *CARF and CADS (R2)*. Such dimension is important for evaluating to what extent the adaptation rules are able to modify the UI.

In order to check whether the adaptation rules cover all the possible levels, it is possible to apply an Inspection-based method. As acceptance criterion, all the three levels have to be addressed (i.e. there is at least one adaptation rule addressing each different adaptation granularity).

Priority: High

This criterion is related to the Adaptation Performance and the Appropriateness of the adaptation.

6.1.2 Adaptation Performance

This criterion refers to the computation time required to perform adaptation. It can be also considered as the delay of system's reaction after adaptation. It might have a strong impact on the user experience.

It is possible to measure the time (in seconds) required to perform adaptation or time (in seconds) required before the system responds after having triggered adaptation. The time can be captured by logging systems events.

Since the performance of the adaptation can be highly dependent on the adaptation granularity (e.g. coarse grained granularity will likely imply higher time for performing the associated adaptation). The acceptance criterion is that the time needed for adaptation should never exceed a certain temporal threshold, which depends on the adaptation and application types.

Priority: High

This criterion is related to the usability of the system (low performance might affect usability of the system), Granularity of the adaptation (coarse granularity will affect the entire UI and will likely need more time to be performed) and the Continuity of the adaptation.

Note: we might also consider measuring subjectively the performance of the adaptation, by asking the user to rate whether they found the amount of time needed for adaptation reasonable. Of course, this will be a different interpretation of adaptation performance (and thus, it is not a technical criterion.)

6.1.3 Support for Open/Extensible adaptation at runtime

This criterion aims to evaluate whether it is possible to introduce at runtime new adaptation rules, or the system is not open to support dynamically the addition of new adaptive behaviour.

This can be assessed by directly checking whether mechanism exists able to support the inclusion of new adaptive behaviour in the application, at runtime through an inspection-based method. As acceptance criterion, at least one mechanism for introducing new adaptation rules at runtime should be supported

Priority: Low

This criterion is related to the Appropriateness of the adaptation.

6.1.4 Flexibility in Adaptation

Adaptation approaches may be *adaptive*, *adaptable*, or based on a *mixed-initiative*. Adaptive personalization mechanisms require no explicit requests from the user, while adaptable mechanisms are driven by the user's choices, and mixed-initiative approaches are a mix of both (e.g. the system provides adaptive suggestions to aid the user in adapting the interface). A flexible system should support all the three mechanisms.

Therefore, it is possible to directly check whether the application covers all the possible levels (e.g. inspection-based method). An acceptance criterion for this measure is that the system supports all the three possibilities, namely in the system there is at least one mechanism for each category.

Priority: Medium

This criterion is related to the Controllability and the Predictability of Adaptation

6.1.5 Customisability of the Adaptive Behaviour

The aim of this criterion is to understand the extent of user control over the adaptive process, namely the users' ability to control both the circumstances that lead to triggering adaptation, and how adaptation is actually performed and applied (i.e., can the user disallow, modify or even cancel an adaptation?).

The capability of modifying the behaviour of the adaptation can be achieved through various techniques: by changing the values of parameters that determine changes in the adaptation results, or even by changing or adding the rules that determine adaptation. An example of applying this criterion in the case of a tool supporting customization of desktop-to-vocal Web application adaptation carried out in the project is reported in (Paternò and Sisti, 2011) in which it is possible to modify parameters that have an impact on the structure of the resulting vocal application or on how it is rendered.

In order to measure to what extent the adaptive behaviour can be customised, it is possible to ask the users to experiment the system by performing specific customisation tasks to judge the level of customisation that is supported in the system. Were users able to modify specific adaptations? Did they feel to need further customisations that were not available in the system? Another measure is the percentage of adaptation rules that can be customized, or the percentage of aspects that can be customized.

An acceptance criterion for this measure is the user's capability to modify some adaptive behaviour in the system, which has an impact on the user experience

Priority: Medium

This criterion is related to the Predictability of the adaptation

6.1.6 Cost of Adaptation

This criterion is about the amount of resources spent/needed to support adaptation. We refer to e.g. the various data needed to perform adaptation, as well as the software/hardware resources needed to support the selected architectural solution for supporting the adaptation process. In other terms, this criterion should give an indicative measure of the cost associated with a specific adaptation architecture, to be possibly compared with another one. Another aspect for assessing the cost associated to the adopted solution is its modifiability and maintainability.

A measure for this criterion is to calculate the cost of modifiability and maintainability of the architectural solution selected for adaptation. Another measure is to estimate/calculate the budget spent for supporting this solution. Another view of the cost is from the user side, in terms of cognitive effort required to understand the adaptation and its results

An acceptance criterion would follow the statement: "the benefits from adaptation should justify its costs". Benefits and costs of adaptation have been analysed in (Lavie and Mayer 2010), which proposes an evaluation of the performance of the users in the context of an in-vehicle telematic system according to four different variables: the task to be accomplished, the type of user (young or elderly), the familiarity with the situation and the level of adaptively. The authors concluded that in familiar situations a completely automated adaptation has high benefits, especially for old people. In unfamiliar situations (non-routine tasks), the cognitive cost for understanding a completely automated adaptation is too high, so the benefits are not justified.

Priority: Medium

This criterion is related to the adoption of the system and the intention to use the system

6.2 User-oriented criteria

6.2.1 Predictability of UI adaptation

Users should be able to understand under which circumstances adaptation takes place, what UI parts are going to be adapted and how the adaptation results will appear after adaptation. Thus, the goal of this criterion is to be able to understand how *predictable* is the adaptation ("Can the user anticipate the system's adaptive behaviour?").

It is worth pointing out that predictability of the adaptation is also relevant for user's confidence and trust in the adaptive system. Indeed, according to (Schmidt-Beltz, 2005), if the system behaviour does not comply with user's expectance or it seems inconsistent in its reactions, this will also undermine user's trust in the system. Indeed, adaptive systems automatically and dynamically adapt to context and assumed user preferences. Therefore, since user models are continuously updated on the basis of observations of user interactions, and such models are exploited to infer system behaviour, in theory, the same user request can cause different reactions today and tomorrow. This might cause surprise and confusion, if the user's mental model of the system reactions is not adequate to explain this behaviour, since the system reactions seem inconsistent, with the result that what was meant to ease use can become a usability issue.

We can assess to what extent the user is able to predict the system behaviour through a 1-5 Likert rating scale where respondents rate their level of agreement with respect to predictability (1= very low predictability and 5= very high predictability, *acceptance criterion*= 4). In other terms it measures how the adaptation matches the mental model of the user.

In addition, this criterion could be evaluated by putting users in front of a specific scenario, then describe to them the current context and ask them what behaviour they would expect from the adaptive system in the scenario and context described and check whether it complies with the actual behaviour of the adaptive application.

The priority of this criterion is *high* as unpredictable adaptations can easily reduce a system's usability by e.g. failing to provide users with a feeling of control: a possible cause is that users have difficulty in building an adequate mental model of the system.

This criterion is related to the User satisfaction (predictability can affect user satisfaction/usability), user confidence/trust in the system (predictability can affect it) and user's awareness of UI Adapation

6.2.2 User's Awareness of UI Adaptation

This refers to the awareness of the user regarding the changes in the UI which are due to adaptation.

The goal is to be able to answer the following question: *"Is the user able to realise that a specific change in the UI was caused by adaptation?"* This can be assessed by gathering user's feedback through explicitly asking users about whether they were aware of some adaptation changes during their interaction with the system. By counting the number of changes identified by users while experiencing the system and then comparing this number to the total number of changes expected in the considered session of the interactive application, it is possible to understand to what extent users are aware of the changes due to adaptation.

It is worth pointing out that sometimes adaptation is so subtle that it can go unnoticed, so in some cases users might be unable to report about adaptivity effects. In order to prevent this, objective measures (e.g. log files) should be also used together with subjective user's feedback.

An acceptance criterion for this measure is that users are aware of the majority of the changes due to adaptation.

Priority: Medium

This criterion is related to the predictability and obtrusiveness of the adaptation

6.2.3 Appropriateness of the Adaptation

When the adaptation decision comes up, it can be applied in different ways (e.g., different colours, different layouts...). The aim of this criterion is to understand whether the system selected a good/appropriate adaptation strategy. In other words, the goal is to be able to answer the following questions: "*When the user realises that an adaptation decision has been reached and the consequent actions have been performed, does s/he judge the performed adaptation as appropriate?*, *"Does the user think that the adaptation improves the quality of interaction with the system?"*

This can be detected through a 1-5 Likert rating scale, where respondents rate their level of agreement with respect to appropriateness of the adaptation after having tried the adaptive application itself (4 is the acceptance criterion).

Another manner to assess this criterion could be to compare different versions of the system (having implemented different decisions for adaptation) and then understand which one is the users' preferred one in terms of appropriateness of adaptation decision. Comparing different adaptation solutions (possibly including also non-adaptive behaviour) allows for estimating the effects of adaptation and can be useful to show that the chosen adaptation decision is the most successful.

A further manner to evaluate this criterion is through interviews which are typically conducted after the interviewee used the system. However, since the user can forget important remarks and experiences, a better solution could be to record the subjects while interacting with the system and then interview them afterward, while analysing together the recorded video. In this way it would be easier to discover the reasoning behind user actions; another advantage is that users are able to remember and mention experiences they might otherwise have forgotten.

Priority: High.

This criterion is related to the Comprehensibility of Adaptation and the User's awareness of the adaptation

6.2.4 Timeliness of the Adaptation

Timeliness of the adaptation refers to the application of adaptation in a timely manner (e.g., not too late, not too early) when there is a need to change some aspect of the user interface to better support the user. The goal is to be able to answer the following questions: *"Does the user think that the adaptation occurs at a time when it actually improves the quality of the user interaction with the system?*;*Was necessary/appropriate to intervene with adaptation at that time?*

This criterion can be assessed through a self-reported 1-5 Likert rating scale, where respondents rate their level of agreement with respect to timeliness of experienced adaptations (acceptance criterion: 4).

Priority: High.

This criterion is related to the Usefulness of the adaptation and Appropriateness of the adaptation.

6.2.5 User's Perceived Confidence and Trust in the Adaptation

In order to adopt the adaptive system, the user needs to be confident in the ability of the adaptive system to predict future needs.

In (Schmidt-Belz, 2005) the author states that while investigating user requirements in adaptive systems, it was found that adaptivity, though meant to increase usability, seems to challenge the overall user's confidence in the system. Then, the author analysed the reasons why users trust in adaptive systems might be low. Indeed, not only privacy, but also user control, consistency, and system competence seem to have strong relationships with user confidence/trust in the system. User control is an important factor in usability, and that control is positively correlated to user confidence/trust in the system, i.e. users trust more when they feel in control. Therefore, in (Schmidt-Beltz, 2005), the author identified trust issues related to pro-active behaviour, which is often included in adaptive systems. Moreover, as it has been highlighted before, still according to (Schmidt-Beltz, 2005), if the system behaviour does not comply with user's expectance and/or seems inconsistent in its reactions, this will undermine user trust.

This can be detected through a 1-5 Likert rating scale, where respondents rate their level of agreement with respect to confidence in the adaptive system (acceptance criterion: 4).

Priority: Medium.

This criterion is related to the Controllability (it is an essential factor to improve user's trust/confidence in the adaptive system), Predictability, Consistency and Accuracy.

6.2.6 Consistency of the Adaptation

We identified two types of consistency, depending on whether it refers to adaptation in the same device or in different devices. In both cases, we below analyse whether the adaptation makes the UI design consistent between before and after the adaptation. For instance, these criteria measure whether similar function keys are used to perform similar tasks throughout the application before and after adaptation. In addition, in graphical UI, labels, and choice of colour should be consistent throughout the application before (we assume that the UI is consistent before adaptation) and after the adaptation

It is also worth pointing out that another interpretation for the consistency of adaptation is that the adaptive behaviour (e.g. in terms of adaptation rules) is applied in a consistent way throughout the interactive system.

6.2.6.1 *Within-Device Consistency of UI Adaptation*

This criterion refers to the consistency of the UI design *after* adaptation with the design *before* adaptation, considering the same device. Therefore, with this criterion we want to assess whether the adaptation is not making the user experience too inconsistent for the user. In order to assess the consistency of the adaptive system we can consider both subjective and objective measures. Among the objective measures, we can consider e.g. the stability of the layout (before and after adaptation). Between the subjective measures, we can ask users to rate it based on a given scale (e.g. 1-5 Likert scale with 4 as acceptance rate) to evaluate to what extent users perceived a consistent adaptation. The measures used are subjective

Priority: Medium.

This criterion is related to the end user disruption.

6.2.6.2 *Across-Device Consistency of UI Adaptation*

This criterion refers to the level of consistency between the UI design *before* an adaptation and *after* an adaptation *to a different device*. An example is when desktop-to-mobile adaptation is performed to support access of a Web application through a device with limited resources. Some level of consistency on the mobile adapted version has to be maintained in order to avoid an excessive cognitive effort for understanding the new interaction model. A manner to evaluate it could be to identify a set of relevant features for comparing user interfaces renderings across different platforms in order to understand their similarity. Another method could be to directly ask user explicit questions by comparing two different interfaces and then ask which one of the two interfaces looks more like the reference rendering, and then derive rough estimates of these parameters by eliciting responses from a significant number of users in a controlled study.

Priority: Medium.

This criterion is related to the Comprehensibility of the adaptation.

6.2.7 Continuity in the Adaptation (technical and subjective)

This criterion refers to the possibility to easily continue the interaction after adaptation.

It can be considered from two points of view. On the one hand, this criterion can refer to the actual, *technical* support provided by the system to continue the interaction after adaptation. For instance, when changing device after adaptation, the system can enable users to continue the interaction on the second device, because it offers the functionality to migrate the state between two devices. However, due to some other contextual conditions or some properties of the adaptation carried out (e.g. a latency time very high), the user might not perceive this adaptation as a continuum, from the point of view of the interaction. Thus, on the other hand, beyond the technical support offered by the system to continue the interaction, we could also evaluate the user’s perceived feeling of continuously interacting with the system. Therefore, we can have a technical criterion and a *subjective* one.

In this case, we can distinguish the case when adaptation occurs in the same device or cases in which there is a change of device. Continuity can be considered at different levels: at *task* level, which means the ability to continue the task even after the user interface has changed due to adaptation; at *action* level, it refers to the possibility to continue, after adaptation, the previously started physical action.

Therefore, it is possible to assess whether there are mechanisms that support continuity before-after adaptation. Beyond that, assess to what extent the adaptation process is perceived as a continuum. Assess whether there are means in the adaptive system which can provide the feeling of continuity (e.g. animated transitions) between the UI before and after adaptation, thus reducing the end user’s cognitive disruption.

Priority: High.

This criterion is related to the Adaptation performance

6.2.8 Transition of Adaptation

This criterion refers to the behaviour of the user interface *during* the adaptation. The adaptation process should be understandable and should allow users to realize what is happening. The goal of the adaptation transition is to highlight that the adaptation process is taking place and provide indications about what is changing. Indeed, in order to reduce the user disruption due to adaptation, the approaching adaptation should be clearly conveyed to the end user. One manner to do this is by e.g. using animated transition scenarios (Dessart, Genaro Motti, and Vanderdonckt, 2011) showing the evolution from the user interface before adaptation to the user interface after adaptation.

In order to measure the user perception, it is possible to evaluate with the users the appropriateness of the representation of the transition, according to the kind of adaptation performed. We can use a 1-5 Likert scale (asking directly users about the perceived usefulness of the adaptive system they experienced) and then the *success criterion* is (reaching at least) 4 in the 1-5 Likert scale.

The Acceptance criterion is that users perceive and appreciate how the transition is represented

Priority: Low.

The criterion is related to the User's awareness of the adaptation and the Continuity of the adaptation.

6.2.9 Accuracy of the Adaptation

This criterion refers to the accuracy with which the adaptive algorithm can predict the user's needs. This is relevant because users are generally faster when adaptation accuracy is higher.

A possible way to measure it is the percentage of the relevant aspects that the adaptation engine is able to address, or the user's perception of the accuracy that can be evaluated through questionnaires or interviews. As acceptance criterion, we should aim to reach a good user's appreciation, which can be detected through a 1-5 Likert rating scale, with 4 as acceptance rate.

Priority: High

This criterion is related to the validity of the modelling process and result and appropriateness;

6.2.10 Comprehensibility of the adaptation

It is important that the user of the system understands how and why automatic adaptation decisions are made ("I understand the rule behind the adaptation").

Measuring this criterion should imply to understand whether the users captured the cause of the adaptation (i.e., does the user understand what triggered a particular adaptation?), as well as on what it will have an impact (e.g., what is / will be adapted and in what way?). These questions can be directly asked to users to assess the comprehensibility of the adaptation, through some a 1-5 Likert scale through which users will assess their level of agreement with respect to comprehensibility of adaptation, with 4 as acceptance criterion

Priority: High

The criterion is related to the Adaptation transition; Consistency and Predictability

6.2.11 Obtrusiveness of the Adaptation

With this criterion we want to understand to what extent the adaptation exploits an appropriate level of obtrusiveness depending on the current context (task, environmental conditions, and urgency) and the current type of applied adaptation. The goal is to have adaptations that adapt unobtrusively to the user, so as to increase users' efficiency and decrease frustration. Therefore, with this criterion we want to evaluate i.e., how obtrusive is the application of an adaptation, with respect to the users' main interaction tasks/current context.

Compared to the criterion of end user disruption due to adaptive behaviour we can say that the obtrusiveness evaluates how obtrusive is the adaptation (per-se), while the end user's disruption evaluates the (negative) effect that this adaptation can have on the users.

A way to measure the unobtrusiveness of the adaptation could be directly to ask users about the perceived obtrusiveness of the adaptation. Another way to measure this criterion could be to check whether adaptation will finally stop some ongoing activities, by e.g. covering some important information on the screen (for graphical UI) or forcing the user to do some additional navigation steps to go back to the point where they were before adaptation.

This criterion is related to the Continuity of the adaptation (if the adaptation is obtrusive, it will likely be an obstacle to the user's perceived continuity) and End user disruption (if the adaptation is obtrusive it will likely be felt disruptive for the user).

6.2.12 End-User Disruption Caused by Adaptation

This criterion evaluates the amount of end-user disruption/frustration caused by the adaptive behaviour. Indeed, the proactive behaviour that adaptive systems generally have sometimes can be a concern for users. Pro-active system behaviour means that an external event triggers a system reaction (instead of an explicit user request). Thus, for the users, one of the potential consequences of adaptation is to be interrupted in the goal-attainment process of an activity due to adaptivity, or to get their attention distracted from something more important, etc. With this criterion we want to evaluate this potential issue.

A measure of this could be to observe and count users' behaviour in which subjects show frustration during the interaction with the adaptive system and this frustration can be brought back to adaptation. Alternatively, users can directly be invited to declare how critical was the interrupted task/how frustrating was the experience/how big was the disruption. Users filling a questionnaire aimed to understand the frustrating experiences they had, or users thinking aloud, or evaluators observing users and annotating behaviour showing frustration. The absence of situations in which users declare to not have been interrupted from the main activity due to the adaptation process is an acceptance factor for this criterion.

Priority: High.

This criterion is related to the Predictability of the adaptation; Controllability, Continuity, Adaptation transition; satisfaction

6.2.13 Impact of Adaptation

The following list of criteria is related to the assessment of the effects of the adaptation with the goal of evaluating its "success" (i.e., whether the goal underlying its introduction has been met). Therefore, these criteria can be seen related to some extent to AAL third-order adaptation rules (not yet specified in Serenoa): there are specific adaptive behaviour that can be applied when we want to improve some specific qualities of the interactive system. In the following we specify which aspect of the interactive system we target.

6.2.13.1 Impact of Adaptation on User's awareness of Unused/Advanced/Overall features of the application-Impact of Adaptation on User's performance of infrequent tasks

One aspect of adaptive interaction is the impact that working in an adaptive interface can have on the user's overall awareness of features in the interface. As highlighted in (Findlater and McGrenere, 2008; Findlater and McGrenere, 2010), when an interface adapts itself to make useful features more accessible for a user's current task, there may be a negative impact on the user's awareness of the full set of available features, which, in turn, can make future new tasks more difficult. Thus, this criterion measures the extent to which the user is aware of the whole set of features of an application including those not yet been used, and provides insight into the potential performance trade-off of working in a personalized UI. For example, an adaptive menu may focus the user's attention on a small set of frequently used features, with the drawback that the user may not see (and thus learn about) additional features.

The study described in (Findlater and McGrenere, 2009) shows that despite the performance benefits of a high accuracy adaptive interface, the adaptive support can result in reduced awareness. Thus, *awareness* should be evaluated in conjunction with performance. Moreover, a balance between efficiency and awareness may change in different contexts. For instance, high awareness of advanced features may be more important for applications where users are expected to become experts. In these cases, an adaptive mechanism that could predict new and potentially useful features for this type of users may be beneficial. On the other hand, applications that are used on a less frequent basis or applications that have users with varying levels of expertise could prefer improving efficiency for core tasks rather than awareness of the full feature set.

To evaluate this aspect of adaptive interfaces, the user's level of awareness of the full set of features in the interface. This awareness can be measured through a recognition test of features in the interface. In addition, a measure of the reduced awareness can be a measurable impact on performance when users are asked to complete new tasks.

As acceptance criterion, the user is aware of 75% of the features existing in the system

Priority: Low.

This criterion is related to Efficiency (trade-off between efficiency and user's awareness of unused features) and Learnability of the whole systems (learning new/advanced/unused features)

6.2.13.2 Impact of the Adaptation on user's expected decrease in interaction complexity (or alternatively, increase in user performance)

One expected result of exploiting adaptation is to have a decrease in the interaction complexity of the system. Indeed, the basic idea is that adaptivity should reduce the interaction complexity (e.g. reduce the amount of navigation required to reach relevant items in the UI). In other situations, by using an adaptive system, the user shall be freed from explicitly and in detail expressing goals, specifying parameters that can be automatically derived, or filling in long forms about settings and preferences. Therefore, with this criterion we aim at evaluating to what extent the adaptation is able to decrease the interaction complexity and then have some positive effect on the user's performance.

One way to evaluate this criterion could be to compare the performance of the user with or without the adaptation and then measure if there is an improvement of the performance (in terms of e.g. decrease in the number of navigation steps).

In order to measure such kind of impact, it is possible to consider a number of common tasks and then compare a non-adapted version of the application with the adaptive one to understand whether the non-adapted version implies a number of navigational steps higher than the adapted version or longer task completion time. Calculate then if the navigation steps (for web/graphical applications) decrease. However, when comparing an adaptive system with a non-adaptive version, we should take into account the specific problems that this comparison could bring (e.g. sometimes the adaptivity is an inherent property of the system which makes this comparison difficult or a non-adaptive version of the system simply does not exist), see e.g. for further analysis on this problem (Weibelzahl, 2002).

Another manner could be to subjectively measure it by using a rating scale and asking users to judge whether they find reasonable the amount of interactivity (e.g. in a graphical UI: scrolling) required by the system to perform a certain task.

Priority: High.

This criterion is related to the Performance of the adaptation and the decrease in the complexity interaction.

6.2.13.3 Impact of Adaptation on User Experience

The goal of this criterion is to understand to what extent adaptive behaviour (in terms of e.g. adaptation rules) can be effective with regard to user experience ("I think the adaptation would be effective with respect to i.e. satisfaction, well-being, enjoyment, etc. of the user"). We can assess to what extent the user finds the adaptation as effective for his/her user experience through a 1-5 Likert rating scale where respondents rate their level of agreement with respect to the impact of adaptation on the user experience (1= very low impact and 5= very high impact, *acceptance criterion*= 4).

Priority: It depends on the application domain (i.e. in the game domain the priority will be quite high).

This criterion can be related to user satisfaction of the overall system

6.2.13.4 Impact of Adaptation on Preventing Critical User Errors

Adaptation could have different goals. In some domains, one of them could be the necessity of making user interactions less prone to errors that can have some critical effect. Thus, in this case the goal of adaptation could be to avoid (or at least reduce) the risk that the user can face critical situations.

A measure for evaluating this criterion is to have some user test and compare two versions of the system to understand which one counts a lower number of errors. As the acceptance criterion we can say that no critical error should be done.

Priority: although it also might depend on the specific application domain considered, a high priority can be assigned to this criterion. In addition, the level of priority might also depend on to what extent the user is able to recover after having done a critical user error

This criterion is related to the Appropriateness of adaptation.

6.3 Adaptation-related Criteria connected with Modelling process/results

6.3.1 Ability of defining and executing adaptation that affects the UI at different levels of abstraction

This criterion means that the Serenoa Framework should offer means to design the adaptation logic at different level of abstraction. It is possible to assess this criterion through an inspection of models, languages and tools used in order to define the logic.

6.3.2 Multi-platform adaptive application design and execution

This criterion means that the Serenoa Framework should offer means to design multi-platform adaptive applications. It is possible to assess this criterion through an inspection of models, languages and tools used in order to define the logic.

6.3.3 Validity of the Context Model

Adaptivity means that the system is automatically adapting its behaviour according to its assumptions about e.g. the current context of use. Then, in order to handle adaptation, the current context has to be observed, modelled/represented so as to dynamically update this model at runtime, i.e. during the interaction. Therefore, this criterion evaluates the validity of the context model exploited by the adaptation. Questions that this criterion should answer are the following ones: *"Is the modelling of the current state of the world correct, based on observations/detections the system gets from the current state of the world?" "Does the user/context model actually reflect the real world?" "Are the interpretations/inferences done by the system valid?"*

Indeed, in order to properly work, adaptive systems have to observe the user behaviour and the conditions of the surrounding context. Thus, the aim of this criterion is to understand whether this data collection process actually works, and whether the user behaviour is accurately registered. In other words, we have to check if the collected data are reliable and the process of collecting those data was sufficiently accurate

A measure for this criterion is the percentage of the contextual aspect that the considered context manager is able to handle correctly.

Priority: High.

This criterion is related to the Appropriateness of the adaptation (if the information contained in the model is not valid, an inappropriate adaptation decision might be taken by the adaptation process) and the Predictability of the adaptation (if the information contained in the model is not valid, the adaptation will likely be unpredictable for the end users).

6.3.4 Coverage/Expressiveness of the context model exploited by Adaptation

With this criterion we want to evaluate the coverage of the aspects modelled and represented in the context model. In other words, with this criterion we are evaluating whether the dimensions used to represent the context are sufficiently comprehensive to specify the most relevant aspects of a context. These aspects will then be considered and exploited in the adaptation process.

Typical questions that this criterion aims to answer are the following: *"Is the world modelled in an appropriate way?" "Are the context characteristics being successfully modelled and stored in the context* *model?"*

A way to measure this criterion is the percentage of context aspects that are modelled.

Priority: Low

This criterion is related to the Validity of the context model. The difference between this criterion and the criterion described before (namely: validity of the context model) is in the fact that the previous one is aimed at checking whether the information contained in the context model is actually valid (i.e. it reflects the current state of the world), while this criterion wants to evaluate whether the context model is expressive enough to be able to represent the most relevant aspects of the context of use.

6.3.5 Scrutability/Transparency of the models

In some adaptive systems, the end users are able to see and understand the meaning of personal information that the system holds about them in e.g. user models. If direct interaction with these models is possible for the users, then we can test so-called *scrutability* (Paramythis, Weibelzhal and Masthoff, 2010). Thus, with this criterion we want to assess the possibility of end-users to correct errors in the exploited models. Therefore we refer to adaptive systems which enable users to review/modify the models maintained (e.g. suppose that there is an explicit UI for modifying models), so that they better reflect their characteristics.

So, in this case, typical aspects we want to understand with this criterion are whether users know what the system has interpreted and consequently modelled about them and the context of use, whether users have understood how the modelling works, whether the users can modify the result of a modelling action

Away to measure this criterion is to check whether there exist e.g. some dedicated UI to allow end users to directly inspect (and possibly modify) the information currently contained in e.g. user's models, so that they can understand and assess to what extent this information is correct.

As acceptance criterion we have that the end users are able to find and edit any property which is found as currently modelled as wrong.

Priority: Low.

This criterion is related to the Predictability of the adaptation.

7 Conclusions

7.1 Summary

This deliverable provides a revised set of criteria useful to assess the results of the projects during their life cycle. Such project results include authoring tools, run-time tools for adaptation and adaptive applications.

We have structured them in terms of technical and user-oriented evaluation criteria. We have indicated a set of assessment criteria more specific to adaptation. We have also discussed how to address user tests to evaluate model-based development of multi-device user interfaces and adaptive user interfaces.

Other deliverables (D5.3.1 and D5.3.2) will show how such criteria have actually been considered in the evaluation of the components of the Serenoa framework and related applications.

8 References

Aggarwal, K.K.; Singh, Y.; Chhabra, J.K.; , "An integrated measure of software maintainability," *Reliability and Maintainability Symposium, 2002. Proceedings. Annual* , vol., no., pp.235-241, 2002

Aquino, Vanderdonckt, Condori-Fernández, Dieste, Pastor, Usability Evaluation of Multi-Device/Platform User Interfaces Generated by Model-Driven Engineering In Proc. of Empirical Software Engineering and Measurement '10.

Bevan, N., Measuring usability as quality of use, Software Quality Journal, 4, 115-150 (1995)

Brusilovsky, P., Karagiannidis, C., & Sampson, D. (2004). Layered evaluation of adaptive learning systems. International Journal of Continuing Engineering Education and Lifelong Learning, 14 (4/5), 402 - 421.

Carta, T., Paternò, F., Santana, W., Support for Remote Usability Evaluation of Web Mobile Applications, ACM SIGDOC, Pisa, October 2011

Denis C. and Karsenty L. 2004. Inter-usability of multidevice systems – a conc eptual framework. In Seffah A. and Javahery H. (eds.) Multiple User Interfaces: Cross-Platform Applications and Context-Aware Interfaces. Wiley & Sons

Dessart, C. E., Genaro Motti, V., and Vanderdonckt, J. Showing user interface adaptivity by animated transitions. In Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '11). ACM, New York, USA, pp.95-104. DOI=10.1145/1996461.1996501 http://doi.acm.org/10.1145/1996461.1996501

Findlater, L., and McGrenere, J., Impact of Screen Size on Performance, Awareness, and User Satisfaction With Adaptive Graphical User Interfaces, CHI 2008 Pages 1247-1256

Findlater, L., and McGrenere, Beyond performance: Feature awareness in personalized interfaces, Journal International Journal of Human-Computer Studies , Volume 68 Issue 3, March, 2010, pp. 121-137

Gajos, Czerwinski, Tan and Weld. Exploring the Design Space for Adaptive Graphical User Interfaces In proc. AVI 2006

Gena, C., Methods and techniques for the evaluation of user-adaptive systems, The Knowledge Engineering Review, 2005, Cambridge University Press. Doi:10.1017/S0269888905000299

Grossman, T., Fitzmaurice, G., & Attar, R. (2009). A survey of software learnability: Metrics, methodologies and guidelines. Paper presented at the CHI '09: Proceedings of the 27th International Conference on Human Factors in Computing Systems, Boston, MA, USA. 649-658.

H. R. Hartson, J. C. Castillo, J. T. Kelso, W. C. Neale: Remote Evaluation: The Network as an Extension of the Usability Laboratory. CHI 1996: 228-235.

ISO/IEC 9241-11 - Guidance on Usability.

ISO/IEC 9126 - Software Engineering - Product Quality

T. Lavie and J. Meyer, "Benefits and costs of adaptive user interfaces," International Journal of Human-Computer Studies, vol. 68, no. 8, pp. 508–524, Aug. 2010.

M. Y. Ivory, M. A. Hearst: The state of the art in automating usability evaluation of user interfaces. ACM Comput. Surv. 33(4): 470-516 (2001)

Jimenez Pazmino, P. and Lyons, L., An exploratory study of input modalities for mobile devices used with museum exhibits. Proceedings of the 2011 annual conference on Human factors in computing systems, CHI '11, pp. 895-904

Paramythis, A., Totter, A. & Stephanidis, C. (2001). A Modular Approach to the Evaluation of Adaptive User Interfaces. In Stephan Weibelzahl, David Chin, and Gehard Weber (Eds.) Empirical Evaluation of Adaptive Systems. Proceedings of workshop held at the Eighth International Conference on User Modeling in Sonthofen, Germany, July 13th, 2001, 9-24, Freiburg.

Paramythis, A., Weibelzahl, S., and Masthoff, J., Layered evaluation of interactive adaptive systems: framework and formative methods. In: Journal on User Modeling and User-Adapted Interaction, Volume 20 Issue 5, December 2010, pp. 383-453

Paternò, F., Sisti, C., Model-Based Customizable Adaptation of Web Applications for Vocal Browsing, ACM SIGDOC, Pisa, October 2011

Rubin, J., Chisnell, D. Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests, Wiley, 2008.

Schmidt-Beltz, B., User Trust in Adaptive Systems. LWA 2005: 69-73.

Seffah, A., Donyaee, M., Kline, R.B., Padda. H.K.,Usability measurement and metrics: A consolidated model. Software Quality Journal (2006) 14: 159–178.

Souchon, C and Vanderdonckt, J, 2003. A Review of XML-compliant User Interface Description Languages. In the *Proceedings of 10th International Conference on Design, Specification, and Verification of Interactive Systems (DSV-IS 2003),* 377-391. http://www.isys.ucl.ac.be/bchi/publications/2003/Souchon-DSVIS2003.pdf

Tullis, T., and Albert, B. Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics, Morgan Kaufmann, 2008.

Van Velsen, L., Van der Geest, T., Klaassen, R., Steehouder, M., User-centered evaluation of adaptive and adaptable systems: A literature review. The Knowledge Engineering Review Journal. Volume 23 Issue 3, September 2008, pp. 261-281

van Vliet, H. "Software Engineering: Principles and Practice (2nd Edition)" Wiley, 1999.

Weibelzahl S., Evaluation of Adaptive Systems, 2002 available at: http://ubttest.opus.hbz-nrw.de/volltexte/2004/234/pdf/20030428.pdf

Acknowledgements

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

- http://www.serenoa-fp7.eu/glossary-of-terms/