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

Project no. FP7 – ICT – 258030

Deliverable 2.1.1

Context Aware Design Space and Context Aware Reference Framework

Due date of deliverable: 28/02/2011

Actual submission to EC date: 25/02/2011

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

Document Information	
Lead Contractor	UCL
Editor	Vivian Genaro Motti
Revision	20.01.2011
Reviewer 1	CNR-ISTI
Reviewer 2	
Approved by	
Project Officer	Paolo Bresciani / Jorge Gasós

Contributors	
Partner	**Contributors**
UCL	Vivian Genaro Motti

Changes			
Version	**Date**	**Author**	**Comments**
1	24/01/2011	UCL – Vivian Genaro Motti	Basic structure of the document: table of content, initial content based in the Description of work

Executive Summary

The main goal of this deliverable is to present two methodologies that compose the framework for context-aware multidimensional adaptation: the Context-aware Design Space (CADS) and the Context-aware Reference Framework (CARF).

These methods are complementary and represent a unified approach that joins context-information and adaptation concepts, such as dimensions, levels and techniques. They consist in graphical representations that organize the dimensions for adaptation in different abstraction levels and the adaptation techniques.

This deliverable defines the methodologies and also presents the background information required to understand the context and motivation for the development and application of them, as well as examples of use and applications of these methods.

Table of Contents

1 Introduction

1.1 Objectives

This deliverable describes the CADS and the CARF: two important components to the development of the framework for adaptation of services front-end according to context awareness. CADS stands for context-aware design space, a graphical representation that provides an overview of the possible dimensions for adaptation and their respective abstraction levels. CARF consists of a context-aware reference framework that refines and details the dimensions of the CADS. An extensive investigation was performed in order to identify and describe in details the adaptation elements and the related techniques. Both methodologies are complementary and provide the main components for the framework aimed by the Serenoa Project.

1.2 Audience

The target audience consists of researchers and practitioners.

1.3 Organization of this document

Chapter 1 presents the goal, audience and related documentation with this deliverable. In Chapter 2, the description of the work is presented. Chapter 3 describes and exemplifies the Design Space. In Chapter 4, the Reference Framework is defined and illustrated with examples. Chapter 5 presents final remarks and concludes this deliverable. In Annex the adaptation techniques gathered so far are described in details.

2 Description of Work

2.1 Motivation

Adaptive system can improve the user interaction in many application domains, for instance students in an e-learning platform, drivers interacting with a GPS, elderly users with a mobile phone. For different situations, different use cases and variables must be taken into account to provide an efficient interaction.

Most of the systems developed are constrained to one specific context of use (an able bodied user, a stable environment, and a desktop PC). However, in reality, the contexts of use vary and can be adapted concerning many different aspects. In order to provide adaptation, the context information has to be identified and considered, mainly it includes (but is not limited to): user, platform and environment. Besides the context information, there are also many dimensions that can be subject to adaptation and in different levels. Previous works started to investigate these concepts, they tried to create, identify and organize adaptation methods, techniques, tools and applications.

However they are constrained to one specific dimension at a time, and they require constant updates. In order to organize adaptation according to their techniques and context information, and to advance previous researches, in the Serenoa Project we define and develop two methods: the context-aware design space and the context-aware reference framework.

The Design Space and the Reference Framework are complementary methods that, together with the Ontology, compose the main framework for the Serenoa Project.

The next sections describe and exemplify both methods.

2.2 Goal

In the context of Serenoa to devise a computational framework for (multidimensional) context-aware adaptation of SFEs three techniques will be used:

- a **Reference Framework** (CARF) that identifies the relevant abstraction levels for the description of SFEs sensitive to the multiple dimensions of the *context of use*.
- a **Design Space** (CADS) that identifies the relevant design options and how they can vary to accommodate different scenarios and requirements.
- an **Ontology** (CARFO) for Multi-Dimensional Adaptation of SFEs

The CARF and the CADS provide a theoretic and conceptual perspective as a unifying approach. We consider adaptations that take into account different contextual aspects (and their mutual influences) at the same time (multi-dimensional adaptation), thus leveraging existing work on adaptive and multi-target UIs.

2.3 Description

Task 2.1 Reference Framework (CARF) and Design Space (CADS) for CAA of SFEs [led by **UCL**]

This task is devoted to devise a sound definition of two main components of the Computational Framework and their relationships:

1. A **Reference Framework** for CAA of SFEs. The starting point to produce the CARF is the Cameleon Reference Framework, which is being leveraged to take into account new technological developments such as Rich Internet Applications (Web 2.0) and Distributed User Interfaces. For this purpose, a conceptual map defining dimensions for context-aware adaptation of SFEs will be progressively built along the following axes: how, when, what, with respect to what, where, etc. This CARF will be fed by already existing surveys on adaptation and will progressively evolve by considering more advanced adaptation techniques. The conceptual map will also be implemented as visualisation software based on hyperbolic trees in order to check consistency, to reason about it, etc.

2. A **Design Space** for multi-dimensional CAA of SFEs: The CADS will also introduce the basic principles that drive context-aware SFEs, enabling to explore different adaptation methods that will be developed during the project. The starting point to produce the CADS will be the FP 6 Similar Adaptation Space that will be expanded and refined.

3 Description of CADS (Context aware Design Space)

The four main goals of a CADS are:

- **introduce the basic aspects** that drive context-aware SFEs.
- **identify the multi-dimensional space** of the solutions for context-aware adaptation of SFEs
- introduce the **architectural layers** that will be responsible for conducting the adaptation process
- be an instrument for **exploring different combinations of adaptation methods** that will be developed during the project

The CADS intends to **support the whole adaptation life-cycle** by allowing each step being performed by the user, the system, a third party or any combination of them. In addition the CADS can consider **feedback or adaptation preferences expressed by end-users or domain experts**, thus enabling an adaptation system that it is capable to **learn** from human beings.

In accordance with the CADS devised, the CARFO **Ontology will be populated** (in the form of *an Adaptation Knowledge Base*) by gathering knowledge about adaptation coming, from different domain experts, existing open-source tools or communities of end-users (for instance, the accessibility community).

3.1 The Similar Adaptation Space

The design space expresses important dimensions and their respective levels in which adaptation must be considered. In an initial version, illustrated by Figure 1, from 2005, the method accommodated 7 context dimensions: *with respect to what*, *who*, *when*, *how many*, *what*, *with what* and *for what*. The design space consisted of two main parts, the upper part of the graphic represented the **action** (regarding the status of the system before adaptation) and the bottom represented the **reaction** (considering the status of the system after the adaptation was performed).

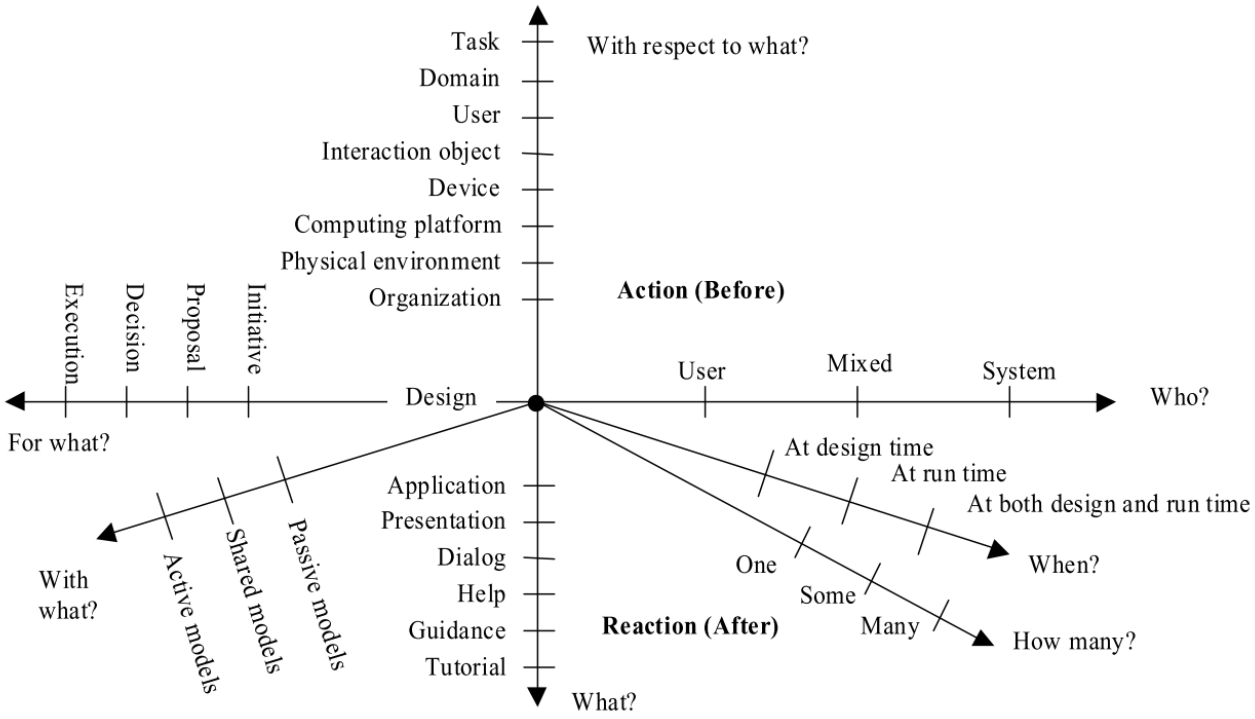


Figure 1. A design space for context sensitivity [Vanderdonckt et al., 2005].

This design space was an initial attempt to graphically organize UI categories for adaptation. The goal of a design space is to help designers to clearly locate, identify and separate the events that cause change of

context as well as the possible reconfigurations of the UI. This attempt was considered limited and was improved in a new version in order to accommodate more dimensions and their respective adaptation levels.

3.2 Context-Aware Design Space

Figure 2 synthesizes the Similar Adaptation Space [Vanderdonckt et al., 2007] for context-sensitive user interfaces. This adaptation space is a result from the efforts of **33 members in 16 countries of the FP6 Similar Network of Excellence** on multimodal user interfaces. The problem space is structured according (but not restricted) to the following dimensions:

- The *means used for adaptation*: re-distribution, re-molding.
- The *UI component granularity* representing the smallest UI units that can be adapted by the way of these means.
- The *state recovery granularity* after adaptation has occurred (from the session level to the user's last action).
- The *UI deployment* (static or dynamic) as a way to characterize how much adaptation has been pre-defined at design-time vs. computed at runtime.
- The *context coverage* to denote the causes for adaptation with which the system is able to cope.
- The *coverage of the technological spaces* as a way to characterize the degree of technical heterogeneity supported by the system.
- The *existence of a meta-UI* to allow users to control and evaluate the adaptation process.

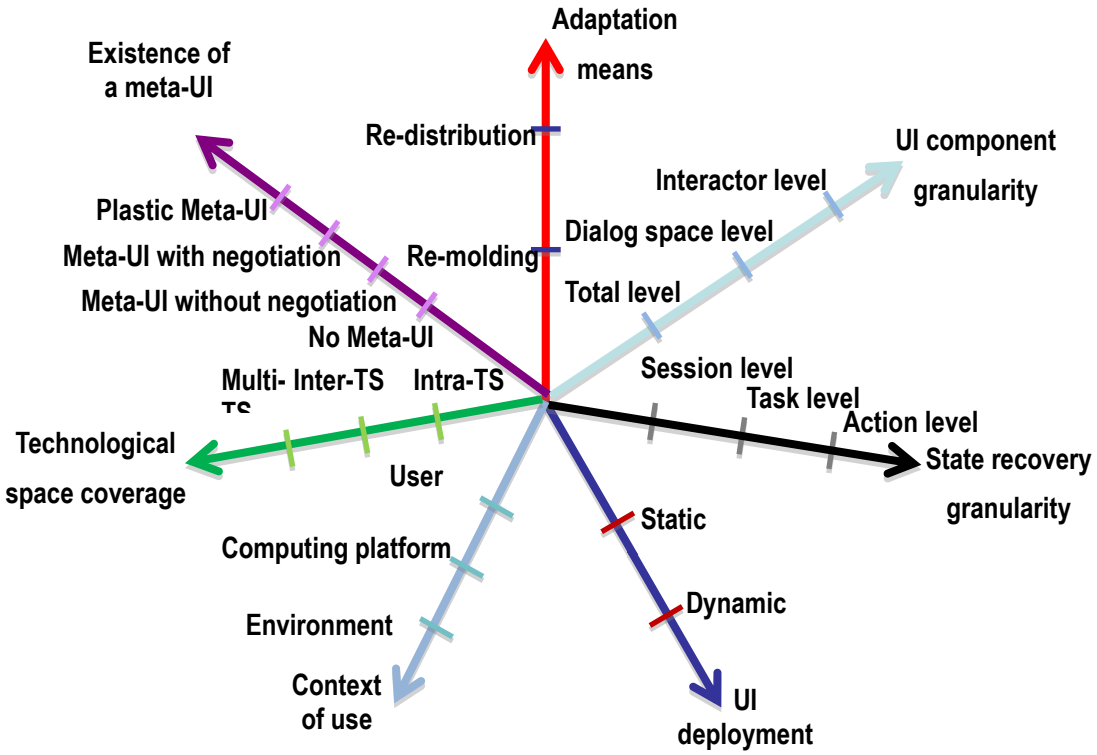


Figure 2. The Similar Adaptation Space [Vanderdonckt et al., 2007].

The dimensions and their specifications that the CADS describes may vary according to different contexts and classifications domains. According to Balme et al. (2004) the platform can be *elementary* (when it is composed of one computer) or a *cluster* (when it is assembled from a set of computers). The assembly may be *static* (if its configuration cannot be modified on the fly) or *dynamic* (when an interaction resource may arrive or disappear). The cluster is *homogeneous* (when it is composed of identical elementary platforms) or *heterogeneous* (the resources and/or the operating system of the constituents differ).

Once the CADS is a flexible and extensible method, it can also accommodate different dimensions. We defined seven that are more important in the context of Serenoa, however other ones can also be considered. These seven dimensions are described and exemplified in the next subsections.

3.2.1 Adaptation Means: Re-molding and Re-distribution

The "Adaptation Means" dimension was extensively investigated concerning its main concepts and techniques. For instance, *UI re-molding* denotes any UI reconfiguration that is perceivable to the user and that results from the application of transformations on the UI. It may result in a *graceful degradation* or in a *graceful upgradation*. Transformations include:

- Insertion of new UI components to provide access to new services relevant in the new *context of use*. For instance, if more screen space becomes available, more information can be displayed.
- Deletion of the UI components that become irrelevant in a new *context of use*. For instance, removing unnecessary UI elements to accommodate screen constraints of a PDA is a frequent technique.
- *Reorganization* of UI components by revisiting their look-and-feel, spatial layout and/or their temporal dependency. For instance, switching from a portrait to a landscape view may require spatial reorganization.

Re-molding may result in using different modalities, or in exploiting multimodality differently. UI adaptation is often assimilated to UI re-molding. This is true as long as we live in a closed world where the interaction resources are limited to a single computer at a time. In ubiquitous computing, the platform may be a dynamic cluster composed of multiple interconnected computing devices whose interaction resources, all together, form a habitat for UI components. In this kind of situation, instead of being centralised, the user interface may be distributed across the interaction resources of the cluster. *UI re-distribution* denotes the re-allocation of the UI components of the system to different interaction resources.

Re-molding a UI from a source to a target UI may imply changes in the set of the supported modalities:

- UI re-molding is *intra-modal* when the source UI components that need to be changed are retargeted within the same modality. Note that if the source user interface is multimodal, then, the target UI is multimodal as well: intra-modal re-molding does not provoke any loss in the modalities set.
- Re-molding is *inter-modal* when the source UI components that need to be changed are retargeted into a different modality. Inter-modal retargeting may engender a modality loss or a modality gain. Thus, a source multimodal UI may be retargeted into a mono-modal UI and conversely, a mono-modal UI may be transformed into a multimodal UI. As for inter-modal re-molding, multimodal re-molding may result in a modality loss or a modality gain.
- Re-molding is multi-modal when it uses a combination of intra and inter-modal transformations. For example, the support of multi-modal re-molding considering graphics and vocal modalities. As for inter-modal re-molding, multi-modal re-molding may result in a modality loss or a modality gain.

UI re-molding may range from cosmetic changes to deep software reorganization.

3.2.2 User Interface Components Granularity

This dimension considers the units of the user interface that can be affected by adaptation. When it is total the whole user interface is affected, but it can also be partial. In this case, the smallest software unit is the interactor. Interactors consist of elements of the interface, such as: input boxes, scroll bars, buttons or dialog windows.

A workspace is considered as the logical space developed to support the execution of a task (or a set of tasks). Workspaces are populated with interactors, such as widgets (graphics, speech), windows, icons and menus. An interface expressed with SVG supports only total remolding, once it is not possible to resize subparts of the user interface to maintain text legibility as the display area is shrunk.

3.2.3 State Recovery Granularity

This dimension characterizes the context in which the adaptation occurred. If the adaptation is applied in the session level, it means that the users must restart their activity from the initial state of the system (when it is launched). At the task level, the user can pursue the activity from the beginning of the task, currently

interrupted. And at the physical action level, the user can carry on the current task at the exact point in which he or she stopped.

For instance, for an email service, the session level corresponds to the act of login in the system, the task level corresponds to send an email, and the action level corresponds to typing the subject of the email message.

3.2.4 User Interface Deployment

The deployment concerns the installation of the user interface component; it may be static or dynamic. When it is static it corresponds to adaptations occurring at the time the system is launched (and after this no changes can be performed). A dynamic deployment allows changes to happen on-the-fly (the use of Ajax technology permits this type of deployment).

Static deployments force user to quit the session and to launch a new and appropriate version of the system.

3.2.5 Context of Use

The context of use was formally defined for this project as the triple: user, environment and platform. Each dimension is a rich information space that includes many characteristics. The environment considers physical and social characteristics.

Dey (2000) defined the context information as all information that is relevant for an application, it concerns all the characteristics of an entity that can be used to adapt a system, for instance, concerning the user, many different aspects can be observed, like the technological experience, gender, age, preferences, language; concerning the environment, we can consider the level of light, noise, humidity, temperature, location; and concerning the platform, characteristics such as, the operating system, memory, battery level, input and output devices, available screens, network connections, bandwidth, and so on.

The context can be gathered in an automated or manual way, by sensors or explicitly requiring the user to set it, for instance.

3.2.6 Technological Space Coverage

The technological space involves a working context and the related: concepts, knowledge, tools, required skills, and possibilities [Kurtev et al., 2002].

Examples of technological spaces include *documentware* concerned with digital documents expressed in XML, *dataware* related to data base systems, *ontologyware*, and so on. Most UIs are implemented within a single Technological Space (TS), such as Tcl/Tk, Swing, or HTML. This homogeneity is not appropriate for plastic multimodal UIs, since re-distribution to different computing devices may require crossing technological spaces. For example, a Java-based UI must be transformed into WML 2.0 when migrating from a PDA to a WAP-enabled mobile phone.

TS coverage denotes the capacity of the underlying infrastructure to support UI plasticity across technological spaces:

- *Intra-TS* corresponds to UIs that are implemented and adapted within a single TS.
- *Inter-TS* corresponds to the situation where the source UI, which is expressed in a single TS, is transformed into a single distinct target TS.
- *Multi-TS* is the flexible situation where the source and/or the target user interfaces are composed of components expressed in distinct technological spaces.

3.2.7 Existence of a Meta-UI

The concept of meta-UI is defined as a simplification of an end-user development environment. A full fledge meta-UI should allow end-users to program (configure, and control) their interactive spaces, to debug (evaluate) them, and to maintain and re-use programs. It binds together the activities that can be performed within an interactive space. In particular, it provides users with the means to configure, control, and evaluate the adaptation process. It may, or may not, negotiate the alternatives for adaptation with the user. It may or may not be plastic.

A *meta-UI without negotiation* makes observable the state of the adaptation process, but does not allow the user to

intervene. The system is autonomous.

A *meta-UI incorporates negotiation* when, for example, it cannot make sound decisions between multiple forms of adaptation, or when the user must fully control the outcome of the process.

The balance between system autonomy and too many negotiation steps is an open question. Another issue is the plasticity of the meta-UI itself since it lives within an evolving habitat. Thus, the recursive dimension of the meta-UI calls for the definition of a native bootstrap meta-UI capable of instantiating the appropriate meta-UI as the system is launched. This is yet another research issue.

3.3 Illustrated Examples for Dimensions' Levels

Many efforts have been done in order to implement adaptive systems. These systems can be adapted in different levels and for specific dimensions. This section illustrates by describing scientific examples, the use of adaptation concerning each dimension of the CADS.

3.3.1 Adaptation Means

Groulax et al. (2005) presented important definitions for the dimension of adaptation means. For instance, in their work they defined the interaction paradigm of *Detachable User Interface* by characterizing it as a set of properties:

- *Detachability:* allows any component of the user interface to be removed from the current interface;
- *Migratability:* allows components of user interfaces to be migrated from a user platform to another (even with distinct operating system, protocols and screen resolution);
- *Plastifiability:* consists in the property of adapt, in an appropriate manner, the interface component that was previously migrated, considering the new context; and
- *Attachability:* consists in a property of merging components to running UI.

The *migration* of a UI was defined as the transference of it from one computing platform to another. It can be considered as *total* or *partial* (depending if the interface is completely or partially migrated); it can be considered as *control-migration* (if only the control component is migrated) or *presentation-migration* (when the control is maintained and the presentation moves), or *mixed* (when both parts are migrated).

In order to accommodate all these properties, a *meta-UI interface* has to be defined: it identifies the possibility of migration, proposes it, selects alternatives, and executes them. Meta-UIs can be *system initiated*, *user-initiated* or *mixed-initiated* (depending on „who" started and performed the migration).

Groulax et al. (2005) also defined UI components as parts of the user interface. These components, that may be subjected to detach, were classified in: *full-screen*, *window*, *active window*, *region*, *fixed region* and a *widget,* according to different abstraction levels. And the computing platform was considered as the *complete hardware and software environment*.

Sedan-Bouillon is a plastic website that can be distributed across a cluster under the user"s control [Balme et al., 2005; Vanderdonckt et al., 2007]. The end user can specify platform screens in which she or he wants to access workspaces that compose the website. Sedan-Bouillon exemplifies the re-molding, as well as re-distribution, they are both performed at the workspace level: the title, content, and the navigation bar are the smallest units for re-molding and re-distribution.

The means used for adaptation are also related with the user interface component granularity, as the next subsection illustrates.

3.3.2 User Interface Component Granularity

User interface component granularity refers to the different unit levels of the interface that can be subjected to adaptation, for instance a window or a button.

Melchior et al. (2009) presented an example in which the interface can be distributed in the interactor-level. Given *n* application processes and m display processes, a toolkit was developed to partitionate the GUI of each application over the *m* display processes that are available at a time. A single display process can, therefore, combine parts of the GUI of each application, in one or several windows with no restriction. This is performed arbitrarily and dynamically.

In this example, as a case study a distributed office suite was implemented. The main goal was to show that a complex application (with many toolbars and buttons) could be distributed over multiple devices and many users. An interface can be decomposed in a lot of components, and a workspace can also be divided into different migratable regions. One of the specific goals was to distribute atomic elements of the user interface, like Figure 3 illustrates. Buttons on the drawing toolbar should be detachable and distributable. A paragraph or a line of the text should also be distributable without the need to distribute the whole text.

The granularity of distribution can range from the application level to the widget level: an entire application can be distributed across platforms, but also the different components of any widget. For instance, even a radio button, consisting of a circle to be checked and a label can be split across platforms. Even the label can be distributed (although it does not make sense in some cases). Buttons on the drawing toolbar can be detachable and distributable.

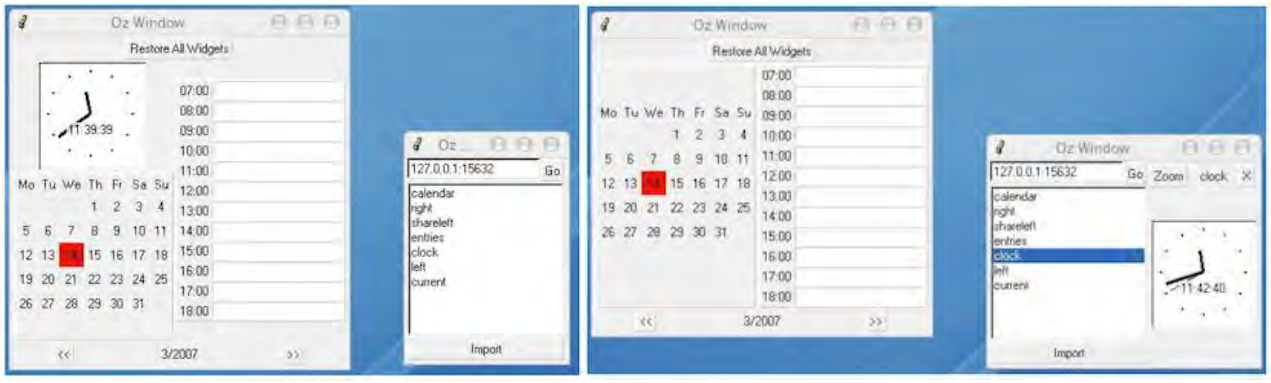


Figure 3. Example of adaptation, the clock is migrated from one application to another [Melchior et al, 2009].

They also considered adaptation for different displays, platforms, operating systems, and users. For instance, each application (parts or the whole) could be marked for exportation, exported to another platform where it continued to run its own life, and re-imported back from where it was initially exported. This migration allows user to access interaction results of other users. Other users can also modify the content created and exported previously. This allows many users to work together no matter where they are located.

3.3.3 State Recovery Granularity

State recovery granularity characterizes the effort users must apply to carry on their activity after adaptation has occurred: at the session level, at the task level or at the action level.

In the example of the Sedan-Bouillon website, in which the plasticity allows it to be distributed across a cluster under the user‟s control [Balme et al., 2005; Vanderdonckt et al., 2007], the end user can specify platform screens in which she or he wants to access workspaces that compose the website. In this case the system supports a task level state recovery: if re-distribution occurs while filling a form, users have to refill the complete form.

3.3.4 User Interface Deployment

A common example of static deployment occurs when the website is able to recognize the use of a specific device, such as a PDA, by, for instance, checking the “user agent” field on the http header, and then the appropriate version of the website is loaded and rendered. This adaptation requires the previous design and implementation of different versions of the website according to different platforms options.

Rekimoto‟s pre-distributed pick and drop is an example of a static UI deployment [Rekimoto, 1997]. Pick and drop consists in an interaction paradigm in which the user selects a resource (a file, for instance) by selecting its icon, and then, the user drags the resource to another available device, in order to make a copy and share it. This paradigm is an extension of the drag-and-drop.

Sedan-Bouillon, on the other hand, supports dynamic UI deployment. The user can select his or her preferences and the interface will automatically change and be adapted according to the choices previously made. There is no need to install a new version of the application in this case, once the website is programmed to provide adaptation on the fly.

3.3.5 Context of Use

Context of use involves all information related to the system. When this context information is relevant it can be used as parameters to compose adaptation rules defined previously.

For Sedan-Bouillon example, a platform is a cluster of connected computing devices. In turn, this cluster may be static or dynamic (computing devices may appear and disappear on the fly), and the cardinality of this cluster is an important factor to consider in relation to scalability and usability. Computing devices are also characterised by the interaction resources they support. The platform dimension of the context of use is probably the most frequently addressed portion of the context to accommodate the UI with.

The context of use was considered in this example concerning: user preferences (where she or he wanted to access each UI component), available devices (PDA and a PC screen) and specific UI components of the application (title, navigation menu and content).

Context information must always be considered for adaptation models, however they consist in a broad range of information, thus the adaptive application available nowadays consider context information, but certainly always in a restricted way. This occurs not only because it is a hard task to gather all possible context information, but also because it is a hard task to infer the best approaches to consider context in a way that adaptation is performed with high usability level and transparency for the users, without confusing or getting them lost.

3.3.6 Technological Space Coverage

Kurtev et al. (2002) asserted that is not easy to define precisely TS, but TS can be considered as working contexts with their associated concepts, knowledge, tools, required skills, and possibilities. TS is often associated to a user community with shared know-how, educational support, common literature and even workshop and conference regular meetings. Examples include the XML TS, the DBMS TS, the abstract syntax TS and the meta-model (OMG/MDA) TS.

Nowadays there are many different technologies available, and though it is hard for developers to decide which is the best solution (of combination of them) to implement an application. Kurtev et al. (2002) provided a higher-level comparative view of 4 technological spaces (XML, MDA, AS – abstract syntax – and Ontologies) according to specific qualities (executability, aspects, formalization, specialization, modularity, traceability, and transformability). These qualities were respectively classified as: poor, fair, good and excellent. The potential interoperability among the TS was also analysed.

The TS as a CADS dimension, can be classified among 3 levels: multi, inter and intra. When the adaptation is performed in the level of *intra* the same TS is maintained, for *inter* the TS changes and *multi* combines different TS. Kurtev et al. (2002) analysed and discussed the issues for this type of adaptation, not always interoperability is provided and sometimes it is necessary to combine processes, such as reverse engineering and forward engineering, for instance to convert Java to C#, first a UML-like model is required.

In the example of Sedan-Bouillon website the technological space is kept even after adaptation is performed, therefore it can be classified as adaptation in the level of intra-technological space.

Technological Spaces can also be classified according to their abstraction levels, for instance a semantic model (such as an Ontology) has a higher abstraction level than a concrete technological space.

3.3.7 Existence of a Meta-UI

A meta-UI incorporates negotiation when, for example, it cannot make sound decisions between multiple forms of adaptation, or when the user must fully control the outcome of the process. This is the case for Sedan-Bouillon where the user can decide on the allocation of the UI components.

Coutaz (2006) proposed the meta-user interface concept as a set of functions that control and evaluate the state of interactive ambient spaces; the goal is to help users in controlling and evaluating this state. A dimension space was presented to classify, compare and contrast research efforts in the domain of meta-UI.

Models and mechanisms are currently being developed for re-molding and re-distribution under the umbrella of plastic UI's and context-aware adaptive UI's. But the level of control left to end-users must be carefully defined.

She designed a prototype for meta-UI that supports the discovery and coupling of mixed-by-design objects (PDA's and PC), as well as UI re-distribution. The integration of the UI of this meta-UI uses a combination of embedded-ness and externality using objects ownership as a driving principle: the user interface of the meta-UI services that act on the UI components of the domain-specific services are embedded.

3.4 Application of the Context-Aware Design Space

The CADS can be applied in the analysis of adaptive systems. A region of the graphic is filled according to the dimensions and levels of adaptation implemented in the system, this approach has two main advantages: it is possible to identify clearly the coverage of adaptation, and consequently, underexplored domains that can be considered to improve adaptation.

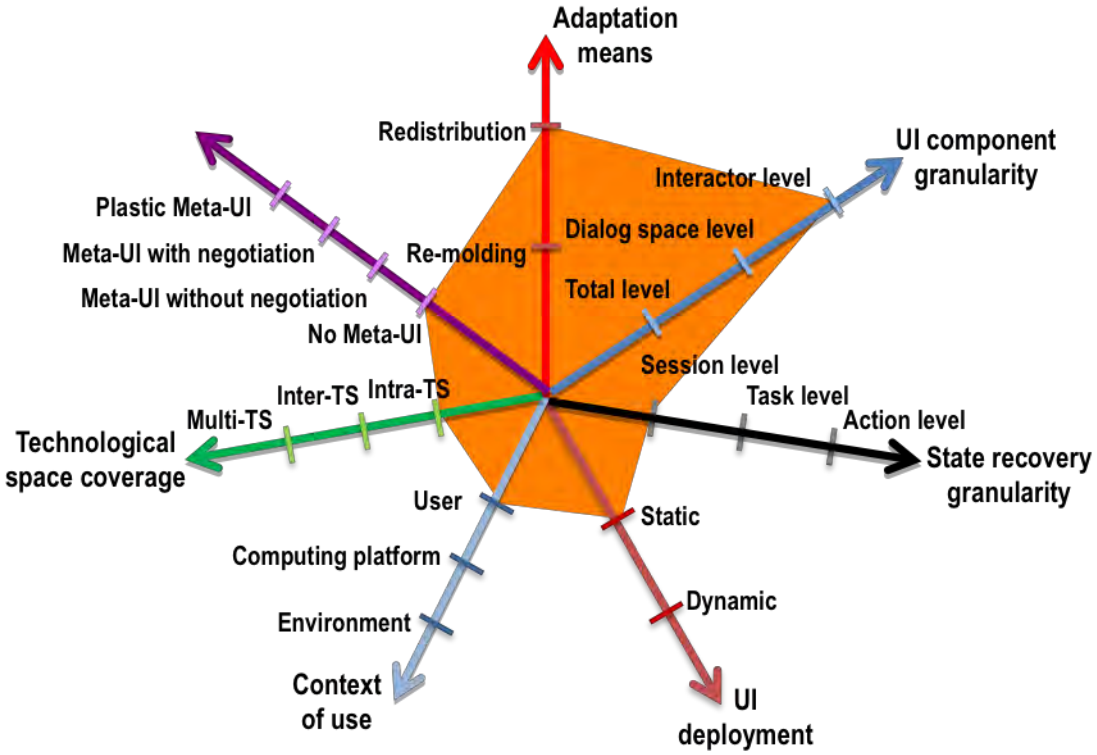


Figure 4. Application example of the CADS for Angie [Schlee & Vanderdonckt, 2004]

Figure 4 illustrates the Design Space applied for Angie, a software implemented aiming design-time adaptability of a GUI by adding and removing UI components and recompiling [Schlee & Vanderdonckt, 2004]. It uses the concept of Generative Programming (GP) that allows automatic creation of softwares using the configuration of elementary and reusable components. In this case, as its CADS illustrates, adaptation is performed by re-distribution; in the interactor level, for instance: radio buttons and check boxes (concerning the UI component granularity); a new session needs to be launched to accommodate the adaptation; the deployment is static; users are being considered for context information; the technological space is maintained (intra-TS) and there is no meta-UI.

The CADS is a method that can be used to classify all adaptive applications. Its use consists in analysing a given adaptive application concerning the dimensions and levels of adaptation provided. Dimension not being provided can be ignored or removed. Besides, the graphic representation of the CADS allows the dimensions to be refined into more detailed levels and new dimensions can be included. The following section provides further descriptions concerning the advantages of the methodology.

3.5 Advantages of the Context-aware Design Space

The CADS is a flexible model once its dimensions can be removed or inserted depending on the application example being analysed. In addition to this, the CADS is an extensible model once it also allows new dimensions to be considered and the current dimensions to be refined accommodating more levels of

adaptation.

The graphical representation of the CADS permits different applications to be compared concerning their adaptation dimensions and levels. In this case the graphic regions considered for two (or more) different applications must be defined, compared and analysed. This approach facilitates the comparison of the adaptation coverage concerning different dimensions and levels. Dimensions not being considered for a given case, or underexplored dimensions, may require special attention.

It is important to take into account the design space goals. According to [Lafon, 2000], a design space should serve three virtues that guide its definition:

- a *descriptive virtue*: any adaptation technique should be described completely, consistently, and unequivocally based on the CADS.
- a *comparative virtue*: each pair of adaptation techniques should be made comparable according to the same criteria defined in the CADS. This supports a sound comparative analysis of techniques, and later on, a rigorous benchmarking of techniques. For instance, if we compare two techniques for CAA of SFEs on the CADS, we will rely on the coverage of the two techniques in order to identify their respective strengths and weaknesses.
- an *exploratory virtue*: all steps of all dimensions of the CADS should be explored in order to identify where are existing techniques, where are new opportunities, and where are under-explored portions of the design space. In particular, we can identify uncovered steps and dimensions of CAA of SFEs, first from an analytic point of view, then according to a technological support.

Therefore in order to consider a new adaptation dimension, its concepts must be initially clarified and well defined. The adaptation levels for a given dimension can be obtained by analysing adaptive applications, previously reported, that provide adaptation for this specific dimension and also by investigating the own nature of the dimension. The description of a new dimension must be complete, consistent and its scope must be well defined.

The comparison between different applications must be done by overlapping the graphical representations of them under the CADS structure. A visual analysis of the outcome provides a clear definition of the adaptation coverage, and consequently the observation of strengths and weakness of the applications being compared. The use of transparency or different colors helps the graphical visualization.

The CADS allows the exploration of adaptive applications concerning all possibilities of dimensions and levels. Its graphical representation permits to identify the coverage of adaptation in a single application, a further investigation of related works provides means to improve and increase the adaptation level in this context.

3.6 Extending the CADS: Further Dimensions

Concerning other potential dimensions of the Similar Adaptation Space, the project participants acknowledge the fact that they have no previous research results to be reused, thus all remaining dimensions will have to be defined and exhaustively investigated during the project lifetime. Nonetheless, due to time and effort limitations we will not deal with all the possible combinations between the different dimensions.

We are elaborating a design space that characterizes the design dimensions of SFE adaptation, based on the initial version of the Similar Adaptation space. This design space is being continuously expanded according to the same mechanisms as for the Reference Framework in order to identify the major dimensions to take into account when performing adaptation of SFEs. Major dimensions to be considered include, but are not limited to: the precise time when the adaptation will be applied (e.g., design time, linking time, compilation time, run-time), the involvement of users, machines and third parties such as a broker (Figure 5), the seven stages of adaptation, the QoSFE (Quality of SFE), etc.

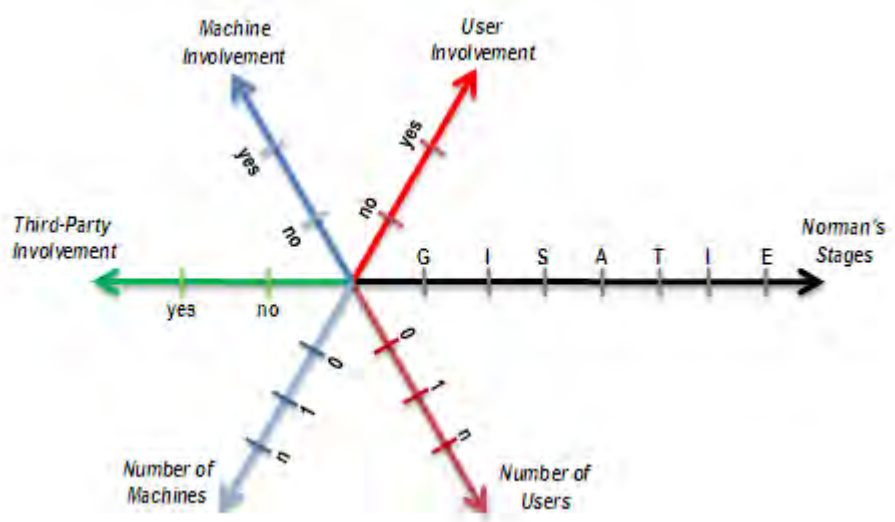


Figure 5. Some examples of other possible dimensions for extending the Design Space.

3.7 Final Remarks

Among the advantages of CADS some must be remarked. This technique is:

- **Extensible:** once the model allows updating, such as the consideration and inclusion of new dimensions;
- **Flexible:** the current dimensions can be removed, replaced or the levels considered for current dimensions can be refined according to the application being analysed or compared;
- **Unified:** the graphic representation accommodates multi-dimensions and different levels in an integrated view;
- **Descriptive:** this technique provides and overview of which dimensions the application provides to adapt itself and in which level it is performed, this allows the analysis of any application;
- **Comparative:** two or more applications can be compared according to the adaptive dimension levels that they provide; it is an advantage to graphically check whether one dimension should be better explored in an application;
- **Exploratory:** many dimensions can be simultaneously considered, designers can classify the application according to its adaptive dimensions and analyse their coverage

The main limitation of the CADS is the difficulty to consider different levels of adaptation in the same axis for comparison. Besides, to compare different applications it is better to consider on top the application that provide a lower adaptation level (in order to provide a better comparative visualization of both cases), the adoption of different line styles (such as hached and continuous line), and distinct transparency levels and colors is also recommended to improve the visualization.

The use of tri-dimensional graphic representation will be investigated once it may help to handle these limitations concerns.

4 Context aware Reference Framework (CARF)

4.1 Background

In the past, many different models with different levels of coverage, expressiveness, and progress have been created and reported. Since recently, the community working in the adaptation domain has produced a general agreement about the abstraction levels appropriate for describing multi-target (i.e. for multiple users, multiple platforms, multiple environments, multiple contexts of use) interactive applications that resulted in the **Cameleon Reference Framework** (CRF) [Cameleon] [Calvary 2003], issued by the FP5 Cameleon project:

The Cameleon Reference framework structures the UI development life cycle into four subsequent layers: *task and domain model*, describing the logical activities necessary to achieve users‟ goals; *abstract user interface (AUI) model*, *concrete user interface (CUI) model*, and *final user interface (FUI)*. The *abstract interface*, is the UI model independent of any target platform and any interaction modality; *concrete interface*, is the UI model which is dependent on the target platform and interaction modality, but independent of any implementation language, and the *final interface*, corresponds to the UI running or interpreted. The contents and usage of these abstraction levels remain however open to interpretation.

Brusilovsky (2001) presented a taxonomy that organizes adaptive hypermedia technologies, however, as illustrates the Figure 6, this taxonomy is limited, once it only considers adaptation regarding presentation and navigation.

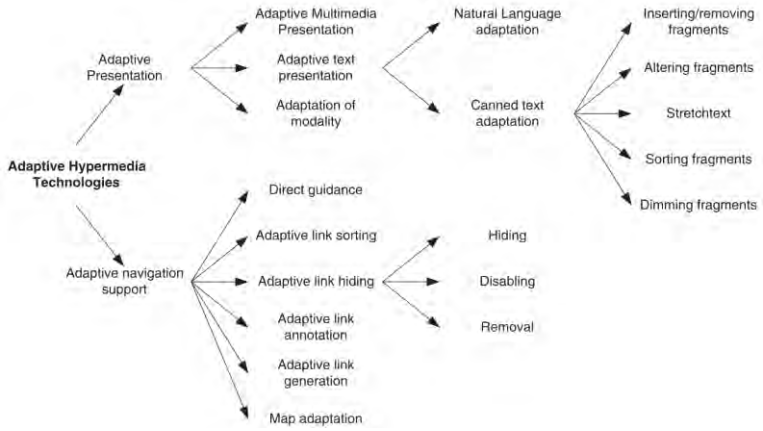


Figure 6. Taxonomy of adaptive hypermedia technologies [Brusilovsky, 2001].

As Bailey (2002) stated the Brusilovsky‟s taxonomy originally provided a mechanism for classifying the various AH systems at the time. Since then, more systems have been developed, some of which fit in to the existing taxonomy, and others that have forced extensions to the taxonomy making it more lengthy and complex.

In 2008, Rouillard proposed a diagram that represents an overview of adaptation. The CARFs implemented so far leverage his work by clarifying techniques for each domain of application. As we can see in Figure 7, this view involves a wide range of context information, but, as the author states, it is not extensive, it represents the main contexts closely related with adaptation.

We extend his model by detailing the techniques applied for each domain, namely content, presentation and navigation. Once Serenoa will consider many different contexts of use for adaptive systems, other domains are also planned to be analysed, such as adaptation techniques for mobile applications.

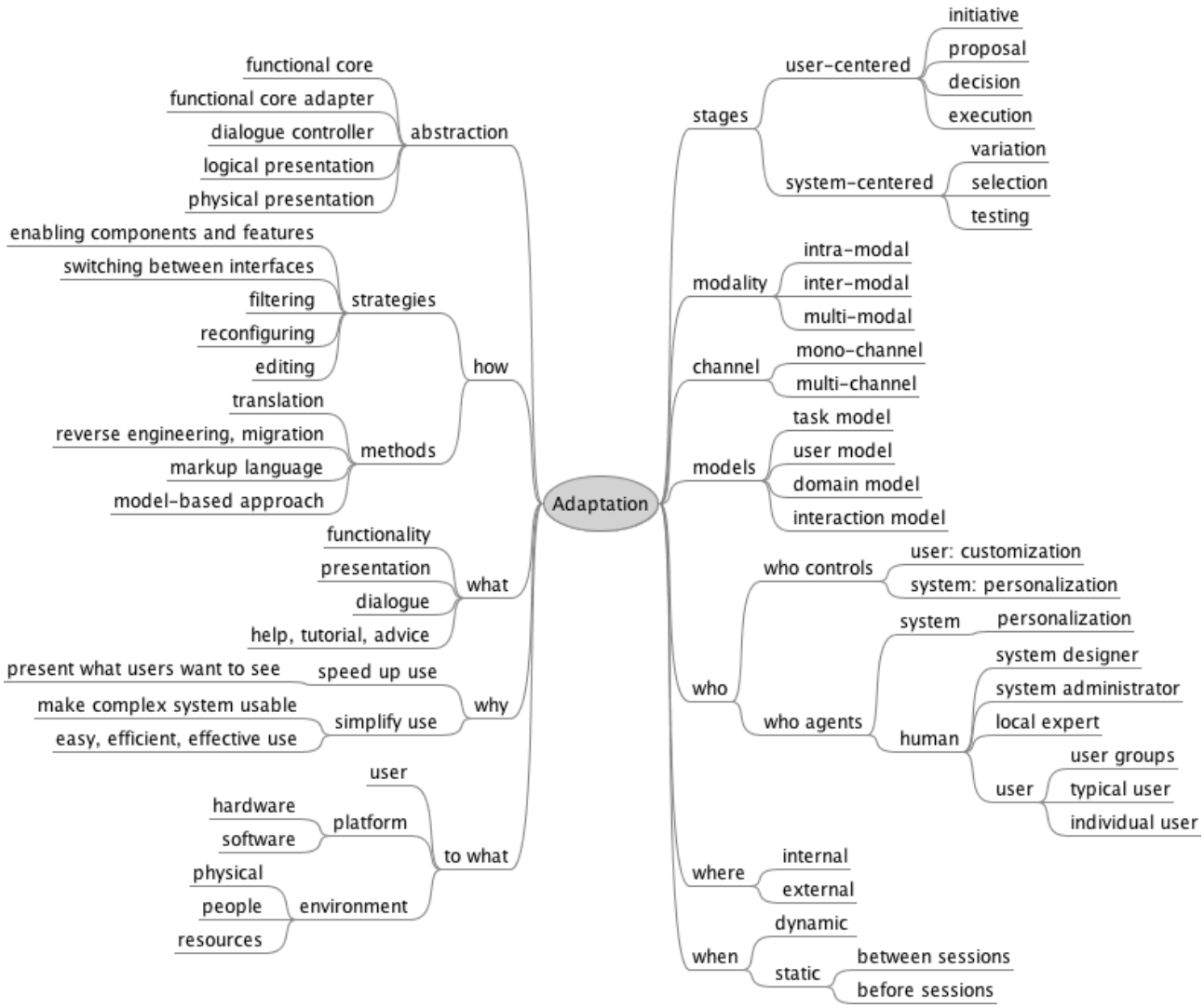


Figure 7. An overview of adaptation based in Rouillard (2008).

4.2 The Reference Framework

In order to provide a rigorous basis for applying the adaptation of SFEs in WPs 3, 4, and 5 in a unified way, there is a need to conduct some fundamental research that systematically analyses existing adaptation techniques. The Computational Framework is addressing this question by researching successively:

- a **Reference Framework** (CARF): is being elaborated in order to articulate all dimensions, their abstraction levels and concepts relevant to adaptation. The content is, in general, being gathered by performing the systematically analysing the literature: by compiling adaptation techniques, abstracting individual techniques into more general ones whether experience permits and consolidating techniques. The techniques are being structured according to how, what, who, when, where, with respect to what and with which models (based on the Cameleon Reference Framework). Finally the techniques are also being generalized when permitted.

Figure 8 depicts a generic representation of the CARF structure according to a tree represented by a mind map. This conceptual map is being consistently explored and defined. It evolves with the consideration of new adaptation techniques and it will be encoded in a system in order to reason about structural differences of adaptation techniques.

Figure 8. Reference Framework

A CARF is being elaborated for each domain of the system subject to adaptation, namely: content (audio, image, text and video), presentation and navigation. The CARF lists all techniques gathered for this domain, and in order to describe them a template was created. This template describes a set of information about each technique listed in the CARF. The template, as Figure 9 illustrates, contains 11 fields that present essential information, such as: its name, description, references, rationale, example, context, advantages, disadvantages, a code sample (e.g. an algorithm), a picture (e.g. snapshot) and additional comments. Currently, a website was created to organize, list and describe techniques, in a systematic way (using a template to detail the techniques gathered).

This template must be used for all techniques presented for this project.

Name:	A short description of the main goal of the adaptation technique.
References:	One or more references that published content about the technique.
Description:	An explanation about the specific goals of this technique, as well as expected requirements, input and output.
Rationale:	The sequence of steps for the execution of the technique.
Example:	A description of a use case.
Context:	In which contexts the technique can be used.
Advantages:	The pros of using it.
Disadvantages:	The cons of using it, as well as related problems.
Sample:	A code sample illustrating the use of it.
Pictures:	A snapshot presenting before and after the application of the technique.
Observation:	For any comments that may appear.

Figure 9. Template for Adaptation Techniques

The templates are being continuously filled according to the progress in the review and analysis of the related literature. The description of the adaptation techniques analysed so far is presented in Annex Sections: Annex A – Adaptation Techniques Templates for Content, Annex B – Adaptation Techniques Templates for Navigation and Annex C – Adaptation Techniques Templates for Presentation.

4.3 Application Examples

The reference frameworks represent one domain of adaptation at a time. For Serenoa, we are considering adaptation for: Content, Navigation and Presentation. Content involves the following formats: audio, image, text and video.

An example of a Reference Framework for text adaptation is illustrated by Figure 10, the related context information consists in the following dimensions: *who*, *what*, *how*, *where* and *why*.

The dimension *what* defines the subject of the adaptation, which is text in this case. The dimension *who* defines the subject that performs the adaptation, the subject can be a person, such as the end-user, or an external agent, like an external application. In this example the text adaptation can be performed by: the system, developer, end-user or a third-party. The dimension *where* defines the device types in which the adaptation occurs, for instance a desktop computer, or a mobile device. The dimension *why* defines the goal of the Reference Framework, thus in this case, adapt to the user. The dimension *how* defines the technique to be applied, thus text processing. *How* can be refined in a sub-level where the different techniques are presented. In this example, the content of the text can be adapted, the content can be: summarized, simplified, truncated, translated and described. And the presentation of the content can be adapted concerning: the zoom (in/out), size, color, font type, orientation, highlight, alignment, spacing, serif,

readability, contrast and background color.

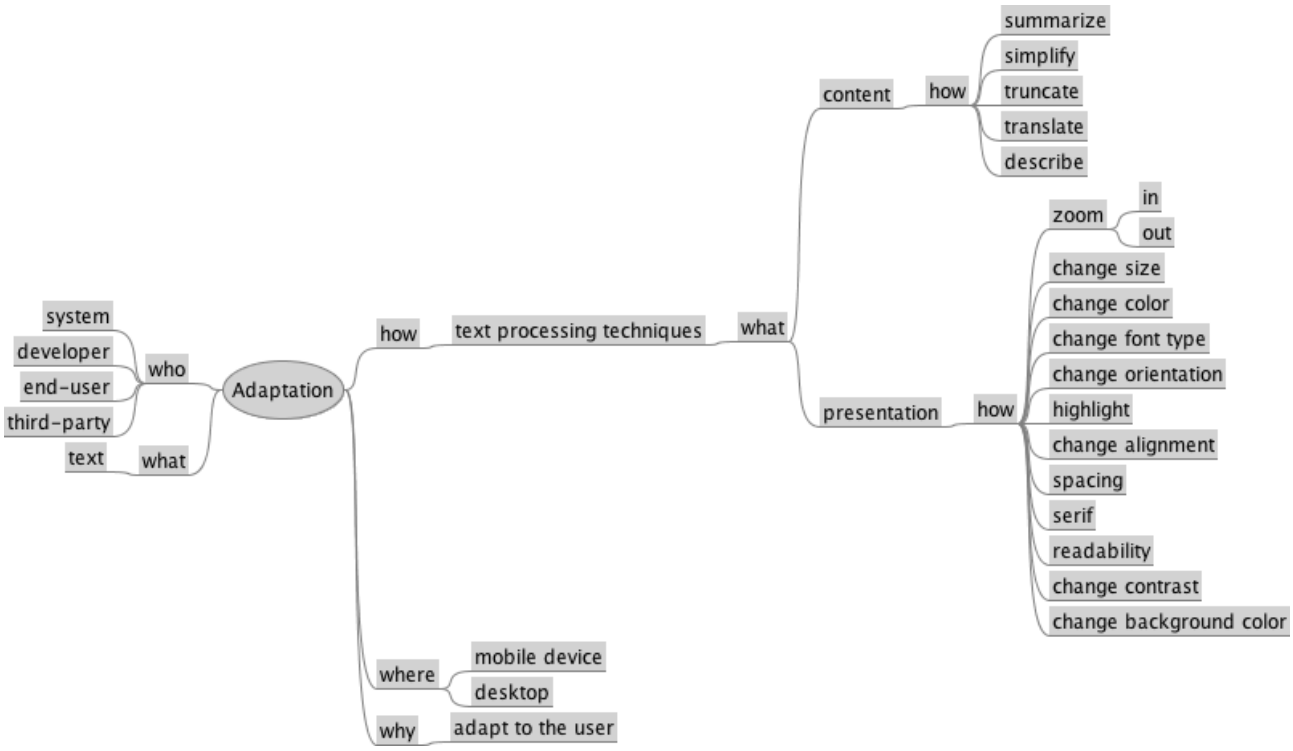


Figure 10. An example of CARF – Adaptations Techniques for Text Content.

The advantages of creating a reference framework for each domain of adaptation are:

- A clear visualization of all possible contexts
- Priorities can be attributed to the contexts to support the composition of adaptation rules
- The model is extensible, once new information can be added
- The model is flexible, information can be collapsed, removed, replaced

However, to achieve a significant coverage of adaptation techniques, an intensive effort is necessary to analyse and consider the works reported so far in this domain. The techniques gathered also need to be refined, once some of them can be grouped, abstracted, refined or extended.

The next sections of this deliverable describe the CARF for the three domains of adaptation according to the analysis of current techniques performed so far.

4.3.1 CARF – Content

The content of an application is usually complex, or composed by different content formats, thus it is possible to adapt the content of an application in a total level, or concerning its different parts (formats). We consider adaptation techniques for the higher abstraction level and also for specific formats that can be used, namely: audios, images, texts and videos. Figure 11 illustrates the CARF for Content Adaptation including the 7 context information previously described. This content represents a sample of all possible options for adaptation concerning the content. This domain will be better explored.

Different techniques have been proposed in order to present information concerning different content types. Annex A – Adaptation Techniques Templates for Content describes in details the three techniques listed, they were defined by Brusilovsky (1997) mainly for text content in web document but they can also be abstracted and applied for other content formats and other domains of application.

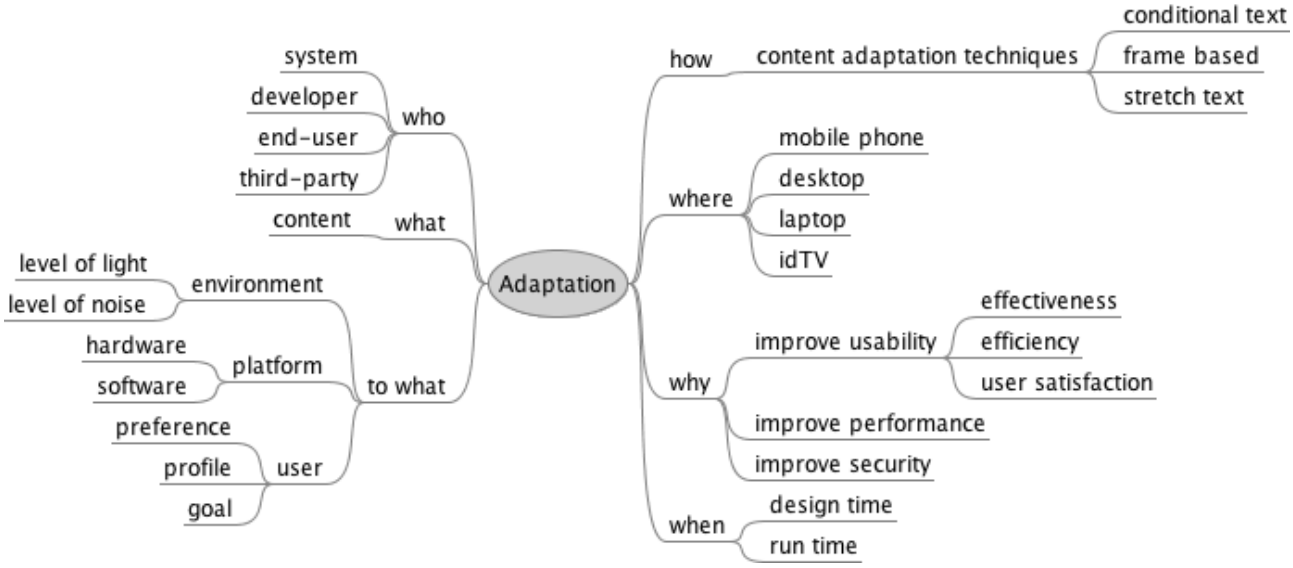


Figure 11. CARF for Content

4.3.1.1 CARF – Audio

Audio content is mainly present in music files or feedback for the user. It is also associated with some specific context, such as with visually impaired users or contexts of use in which a screen is absent or can difficult the main task of the user (e.g. when the user is driving a car). Therefore audio adaptation should especially consider the context of the user, and the environment (task of the user, level of noise).

Some examples of adaptation techniques for this domain are: the reduction of the bit rate of an audio file, the reduction of the rate at which an audio file is sampled, the conversion of an audio file from two to one single channel (in order to reduce the required bandwidth to transfer the audio content), the conversion of the audio format (e.g. WAV to mp3), the translation of the audio content, the translation between modalities (e.g. from audio to text) [Lemma et al., 2005]. The Figure 12 illustrates the CARF concerning audio content.

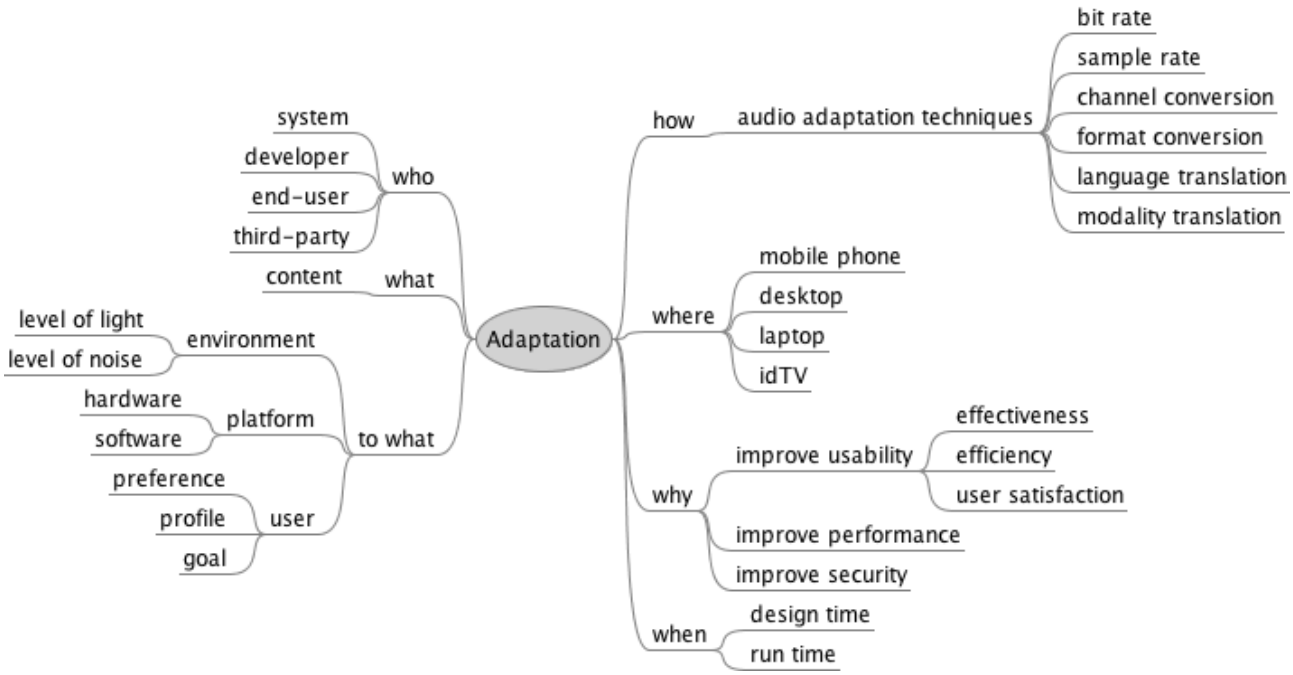


Figure 12. CARF for Audio

These techniques will be better investigated and detailed in future works.

4.3.1.2 CARF – Image

Regarding image content, different techniques have been proposed in order to present graphical information for different contexts of use. Figure 13 illustrates the CARF for image adaptation. It contains 15 techniques that change visual content. These techniques will be better explored and detailed in future works.

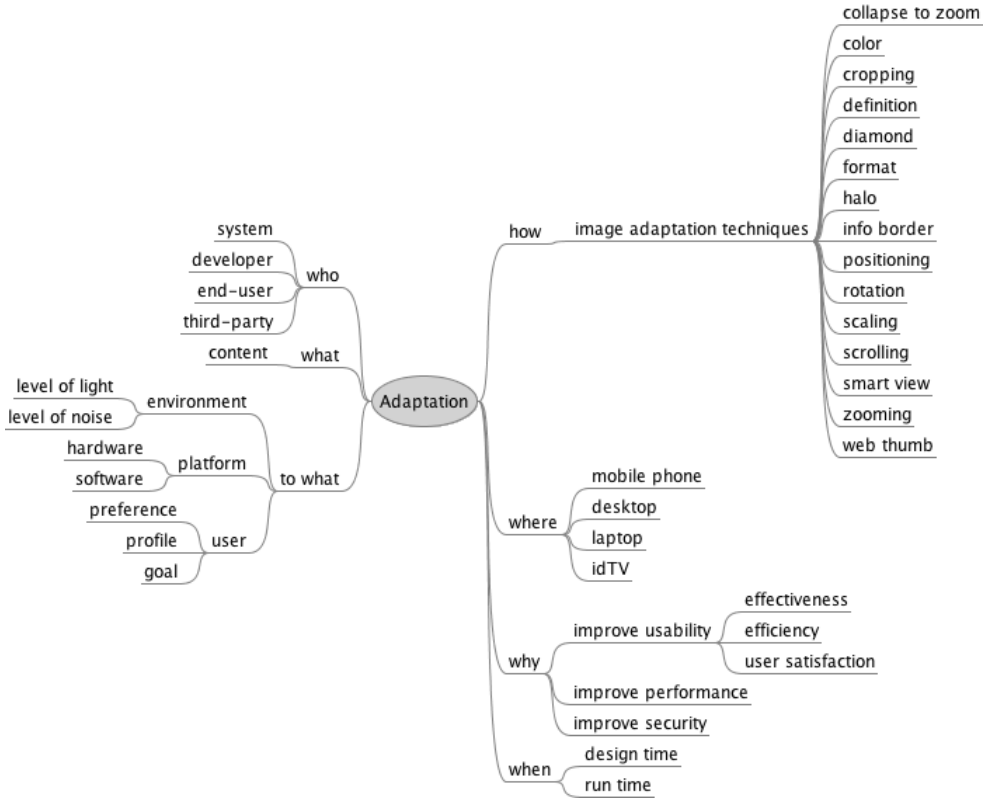


Figure 13. CARF for Image

4.3.1.3 CARF – Text

The text content is probably the most common type of content that can be found in technological applications, when users access a website, interact with a mobile phone, or with a digital television, their means of interaction are mostly based in „written" communication channels (e.g. keyboards, remote controls). Besides there are many features that can change in this content format, its presentation and its content. Therefore there is a wide range of adaptation techniques that can be considered in this domain.

The Figure 14. CARF for Text illustrates the CARF containing 28 adaptation techniques for text content. The detailed description of these techniques can be seen in Annex B – Adaptation Techniques Templates for Content - Text.

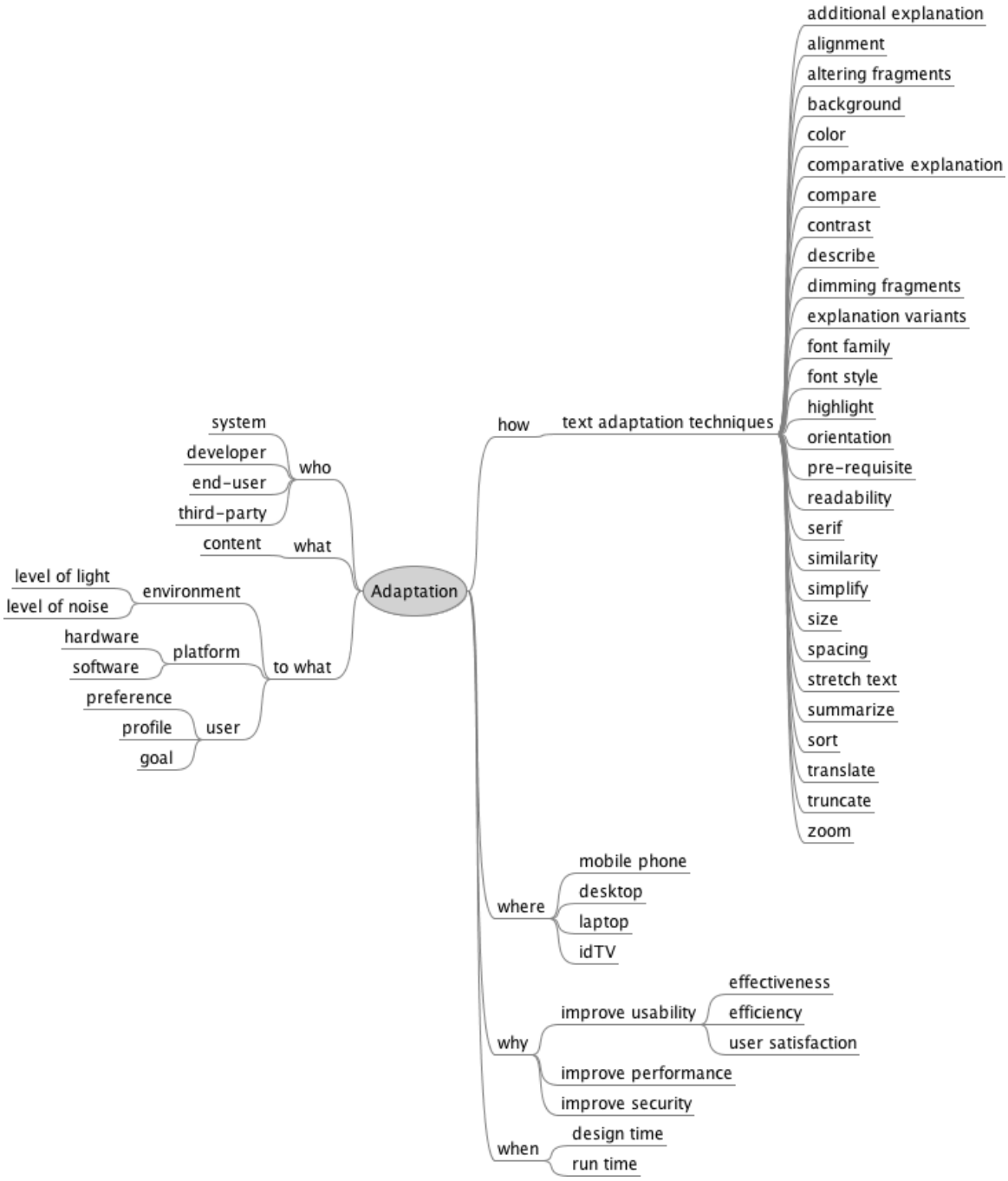


Figure 14. CARF for Text

4.3.1.4 CARF – Video

There are many different adaptation techniques that can be considered for video content. Although video files are mostly used in the domain of interactive digital televisions, they can also be presented in desktop computers, laptops or smaller devices, such as mobile phones or PDAs. Adaptation in this domain can vary according to multiple context information, such as available bandwidth, scalability, video quality, user satisfaction, preferences and so on.

The CARF illustrated by Figure 15 presents 6 adaptation techniques for video content [Cotroneo et al., 2005]. In a future analysis we plan to gather and detail more techniques used in this domain.

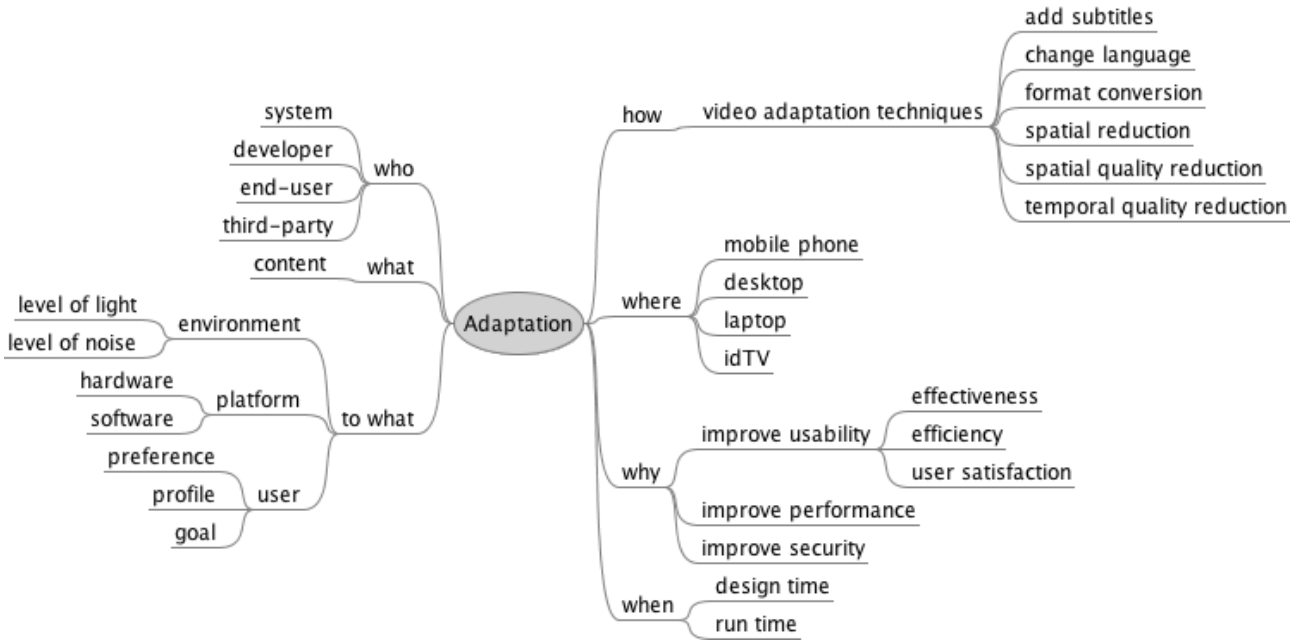


Figure 15. CARF for Video

4.3.2 CARF – Navigation

Different techniques have been proposed in order to adequate the navigation options concerning different contexts. Figure 16 illustrates the CARF for navigation techniques. The reference used was Brusilovsky (1999), in principle 6 different techniques were gathered and described in details, the continuous investigation of related literature will provide other techniques to be added in this CARF. A template for each of these techniques was filled, it describes in details information about them. The templates can be consulted in the Annex B – Adaptation Techniques Templates for Navigation.

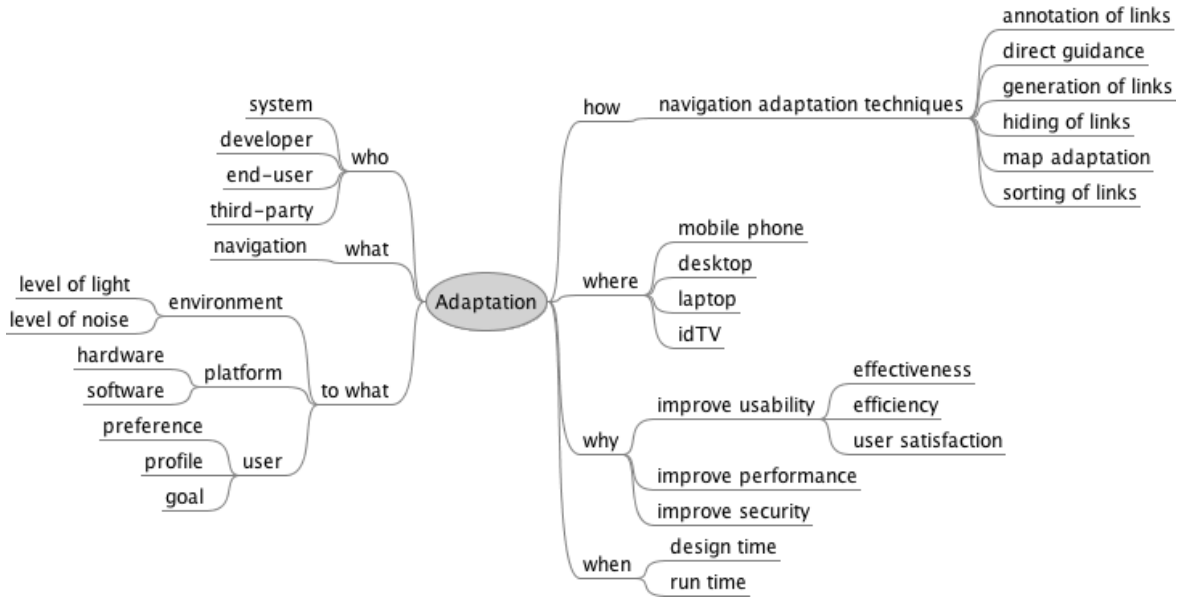


Figure 16. CARF for Navigation

The navigation is defined by structure of the links that the user can follow while interacting with the application. The navigation can be adapted according to user preferences or expertise, for instance interaction options (represented usually by menu items, links or buttons) can be prioritized according to the context of use, more relevant options must be available, less relevant options can be either grouped, collapsed or even removed, this adaptation not only optimises the user interaction but also permits a more clear visualization of

the interaction options that a user have available in the interface in a given moment. The space in a presentation interface may be also saved, what is appropriate for small screen devices or higher levels of zoom, for instance.

It is relevant to consider also interfaces that are not visual, navigation for auditory systems, for example, must also be considered once their navigation can be adapted as well.

4.3.3 CARF – Presentation

The presentation in an application can change for many different reasons, such as: a different screen size, visual impairment of the user, a different position of the output devices (e.g. screens can be in landscape or portrait), a changing in the environment (like the level of light) and so on. Different techniques have been proposed in order to present information with different layouts according to different contexts. Figure 17 illustrates the CARF for presentation. It contains 7 different techniques; and their description can be consulted in the Annex C – Adaptation Techniques Templates for Presentation.

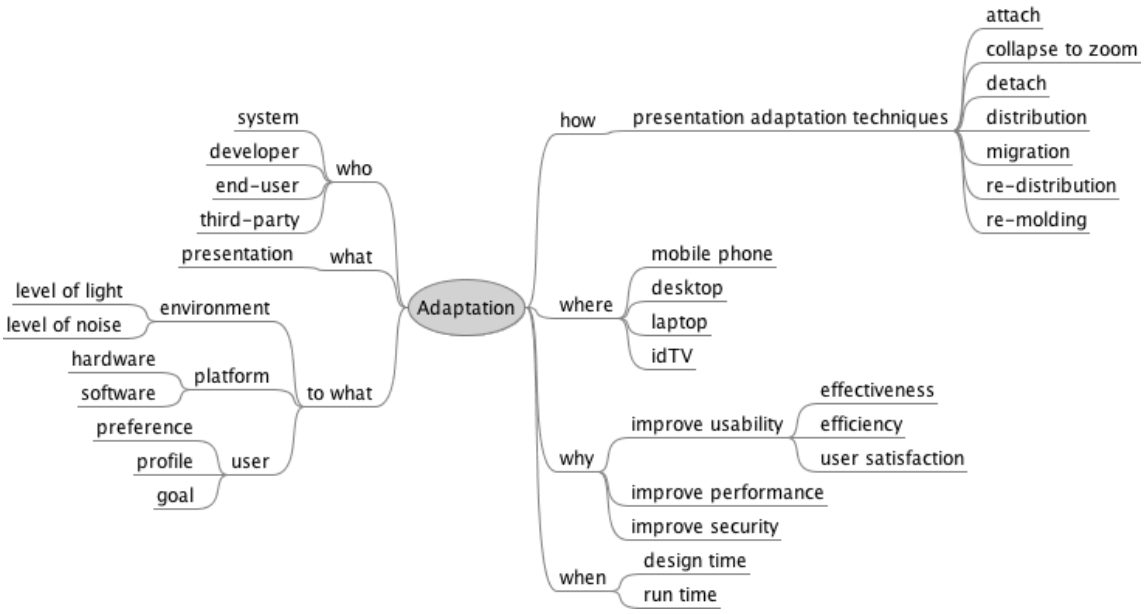


Figure 17. CARF for Presentation

The techniques to adapt presentation will be further investigated in future work in order to identify other techniques, extend the CARF and refine the techniques gathered so far (for instance by grouping similar techniques described in the literature with different names, such as distribution and re-distribution).

4.4 Final Remarks

The CARF methodology is flexible, extensible and unified. Its context can be added, removed, and extended. It is considered unified once all contexts can be seen in a single diagram.

It is a complex task to elaborate a Reference Framework that covers all the possible dimensions for context-aware adaptation, therefore it is mandatory to perform an extensive investigation to gather as much as possible the dimensions for each context domain subject to adaptation. To compose the templates and to elaborate the CARFs an extensive analysis of the literature is being conducted.

The elaboration of CARFs for all adaptation dimensions demands a significant effort, however its outcome will generate a unified model that organizes many different adaptation techniques. This outcome will be used to feed adaptation rules in the next steps of the project. Context information will also be considered to prioritize distinct adaptation techniques.

A continuous and critical analysis of the outcomes is necessary in order to refine the techniques gathered, the adaptation techniques described can be classified, grouped, abstracted, refined and associated, further investigation is planned to achieve a more complete version of them.

5 Conclusion

Together, the CARF, CARFO and CADS will constitute a **computational framework** for the development of a set of new technologies, which will enable a new generation of SFEs sensitive to different aspects of the *context of use* at the same time.

It is important to conclude the CARF and CADS to cover a wide range of dimensions and levels for adaptation. The gathering of theses techniques, as well as a consistent description of them, may support developers in building adaptive applications.

The methodologies will serve as a material of reference, and the adaptation rules can be reused in order to facilitate and promote the development of adaptive systems in several scenarios.

5.1 Final Remarks

These methodologies require an extensive investigation to cover as many aspects as possible for context-aware adaptation. A continuous investigation and refinement is necessary in order to keep the documents updated and complete.

5.2 Future Work

The next efforts will consist in reviewing and analysing continuously the literature, in order to gather more adaptation techniques regarding as many dimensions as possible. The descriptions of the techniques will also be periodically checked and refined in order to avoid inconsistencies and to detect mistakes or repetitive information.

In a first analysis it is possible to observe that the techniques can be classified according to different aspects, one of them is the complexity, *complex* techniques are composed by the combination of other techniques, while *simple* techniques consist in one single transformation. For instance, modifying the readability of a text content consists in applying a set of adaptation techniques, such as changing the: alignment, contrast and size of the text.

The techniques can also be grouped, for instance, regarding the adaptation for text it is possible to observe that some techniques changes the *presentation* and others the *content*. Presentation is modified by alignment, color, contrast, size, and so on; while the content itself is modified by summarizing, truncating, or explaining it, for example.

The work done so far investigated techniques regarding mainly content, navigation and presentation, we started also to analyse techniques according to the platform, such as the adaptation techniques that W3C proposed for mobile devices (http://www.w3.org/TR/di-atdi/).

Once the literature review is done, the association of the techniques with context information (mainly regarding: platform, environment and users) will be further investigated and defined in order to compose adaptation rules that consider both issues (context and techniques).

References

[AHA!] De Bra, P. & Calvi, L. (1998). AHA: a Generic Adaptive Hypermedia System. 2nd Workshop on Adaptive Hypertextand Hypermedia, pp. 1-10. (URL: http://wwwis.win.tue.nl/ah98/DeBra.html)

[Armstrong et al., 1995]
. AAAI Spring Symposium on Information Gathering from Distributed, Heterogeneous Environments, Stanford, CA, http://www.isi.edu/sims/knoblock/sss95/mitchell.ps.

[Bailey, 2002] Bailey, C., Hall, W., Millard, D., and Weal, M.Towards open adaptive hypermedia. In Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002) (Malaga, Spain,May 2002).

[Balme et al., 2004] Balme, L., Demeure, A., Barralon, N., Coutaz, J., & Calvary, G. (2004). CAMELEON-RT: A Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces. *Ambient Intelligence* (Vol. 3295, pp. 291-302). Springer. Retrieved from http://www.springerlink.com/content/p27tmcvqw24r2ljj

[Balme et al., 2005] Balme, L., Demeure, A., Calvary, G., Coutaz, J.: Sedan-Bouillon: A Plastic Web Site. In: PSMD 2005, the INTERACT 2005 Workshop on Plastic Services for Mobile Devices, Rome, Italy (2005).

[Beaumont, 1994] Beaumont, I.: 1994, User modeling in the interactive anatomy tutoring system ANATOM-TUTOR. User Models and User Adapted Interaction 4 (1), 21-45.

[Boyle & Encarnacion, 1994]
. User Models and User Adapted Interaction 4 (1), 1-19.

[Brusilovsky, 1997] P. Brusilovsky, Efficient Techniques for Adaptive Hypermedia, Intelligent Hypertext, pp.12-30, Springer, 1997.

[Brusilovsky et al., 1998] Brusilovsky, P., Eklund, J. and Schwarz, E.: 1998, Web-based education for all: A tool for developing adaptive courseware. ComputerNetworks and ISDN Systems 30(1-7): 291-300, 1998.

[Brusilovsky, 2001] P. Brusilovsky. Adaptive hypermedia. User Modeling and User-Adapted Interaction, 11(1-2):87-110, 2001.

[Brusilovsky, 2003] Brusilovsky, P. (2003), Adaptive navigation support in educational hypermedia: the role of student knowledge level and the case for meta-adaptation. British Journal of Educational Technology, 34: 487–497. doi: 10.1111/1467-8535.00345

[Calvary et al., 2003] Calvary G., Coutaz J., Thevenin D., Limbourg Q., Bouillon L., Vanderdonckt J., 2003. A unifying reference framework for multi-target user interfaces. Interacting With Computers,Vol. 15 No. 3, pp. 289-308.

[Cotroneo et al., 2005] D. Cotroneo , G. Paolillo , C. Pirro , S. Russo, A User-Driven Adaptation Strategy for Mobile Video Streaming Applications, Proceedings of the First International Workshop on Services and Infrastructure for the Ubiquitous and Mobile Internet (SIUMI) (ICDCSW'05), p.338-344, June 06-10, 2005 [doi>10.1109/ICDCSW.2005.28]

[Coutaz, 2006] Coutaz, J. (2006) Meta-User Interfaces for Ambient Spaces. In: Proc. of 4th Int. Workshop on Task Models and Diagrams for User Interface Design Tamodia"2006 (Hasselt, October 23-24, 2006). Lecture Notes in Computer Science, Vol. 4385, Springer, Heidelberg, pp. 1–15.

[Dey, 2000] Dey, A.K., Salber, D. & Abowd, G.D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, Human-Computer Interaction, (pp. 97-166), 16(2, 3, 4).

[ELM-ART] [Brusilovsky, Schwarz & Weber 1996] Brusilovsky, P., Schwarz, E., & Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. In Frasson, C., Gauthier, G., & Lesgold, A. (Ed.), Intelligent Tutoring Systems (Lecture Notes in Computer Science, Vol. 1086). Berlin: Springer Verlag. 261-269.

[Encarnacao, 1995] , L. M.: 1995a, Adaptive help for interactive graphics systems: an application- independent approach. Workshop Adaptiviteit und Benutzermodellierung in interactiven Systemen (ABIS 95), Munchen.

[Eklund & Brusilovsky, 1999] Eklund, J. and Brusilovsky, P. (1999) InterBook: An Adaptive Tutoring System Uniserve Science News Vol. 12. March 1999. p. 8-13.

[Fischer, 1990] - . 23-th Annual Hawaii International Conference on System Sciences, Kailua-Kona, HI, pp. 309-317.

[Gonschorek & Herzorg, 1995] . AI-ED'95, 7th World Conference on Artificial Intelligence in Education, Washington, DC, pp. 274-281.

[Grolaux et al., 2005] Grolaux, D., Vanderdonckt, J., Van Roy, P., Attach me, Detach me, Assemble me like You Work, Proc. of INTERACT‟2005 (Rome, 12-16 September 2005), M.-F. Costabile, F. Paternò (eds.), Lecture Notes in Computer Science, Vol. 3585, Springer-Verlag, Berlin, 2005, pp. 198-212.

[Hohl, Bucker, GunzenhSuser, 1996] Hohl, H., H.-D. Bocker, and R. Gunzenhauser: 1996, Hypadapter: An adaptive hypertext system for exploratory learning and programming. User Models and User Adapted Interaction 6.

[Hook, 1996] Hook, K., J. Karlgren, A. Wern, N. Dahlback, C. G. Jansson, K. Karlgren, and B. Lemaire: 1996, A glass box approach to adaptive hypermedia. User Models and User Adapted Interaction 6 (this issue).

[Hook, 1997] Hook, K.: Evaluating the Utility and Usability of an Adaptive Hypermedia System. In: Proc. of IUI'97, International Conference on Intelligent User Interfaces (1997) 179-18

[HyperBook] Henze, N. and Nejdl, W. (1999). Adaptivity in the KBS hyperbook system. In 2nd Workshop onAdaptive Systems and User Modeling on the WWW, Toronto, Canada.

[InterBook] Eklund, J. and Brusilovsky, P. (1999) InterBook: An Adaptive Tutoring System Uniserve Science News Vol. 12. March 1999. p. 8-13.

[Kay & Kummerfeld, 1994] Kay, J. and R. Kummerfeld: 1994a, Adaptive hypertext for individualised instruction.Workshop on Adaptive Hypertext and Hypermedia at Fourth International Conference on User Modeling, Hyannis, MA, http://www.cs.bgsu.edu/hypertext/adaptive/Kay.html.

[Kobsa et al., 1994] - - . 4-th International Conference on User Modeling, Hyannis, MA, pp. 31-36.

[Kurtev et al., 2002] Kurtev, I., Bézivin, J., and Aksit, M. (2002) Technological Spaces: an Initial Appraisal. CoopIS, DOA'2002 Federated Conferences, Industrial track, Irvine, 2002, http://www.sciences.univ-nantes.fr/lina/atl/www/papers/Position PaperKurtev.pdf

[Kushniruk & Wang, 1994] Kushniruk, A. and H. Wang: 1994, A hypermedia-based educational system with knowledge- based guidance. ED-MEDIA'94 - World conference on educational multimedia and hypermedia, Vancouver, Canada, pp. 335-340.

[Lafon, 2000] Beaudouin-Lafon, M. Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI 2000), ACM Press, New York, NY, 2000, 446-453.

[Lemma et al., 2005] Surafel Lemma, Dawit Bekele and Girma Berhe, Semantic Description of Multimedia Content Adaptation Web services, Proceedings International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIS2E 05), December 10-20, 2005, Technically Co-Sponsored by Institute of Electrical & Electronics Engineers (IEEE) University of Bridgeport.

[Melchior et al., 2009] Melchior, J., Grolaux, D., Vanderdonckt, J., Van Roy, P., A Toolkit for Peer-to-Peer Distributed User Interfaces: Concepts, Implementation, and Applications, Proc. of 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems EICS‟2009 (Pittsburgh, July 15-17, 2009), ACM

Press, New York, 2009, pp. 69-78.

[Patterno et al., 1999] Patterno, F. & Mancini, C. Designing Web Interfaces Adaptable to Different Types of Use. Proceedings of the Workshop Museums and the Web. At: http://www.archimuse.com/mw99/papers/paterno/paterno .html, 1999.

[Patterno et al., 2008] Paternò F. Santoro C. Scorcia A. "Preserving rich user interface state in Web applications across various platforms". Engineering Interactive Systems. Second Conference on Human-Centered Software Engineering, HCSE 2008 TAMODIA 2008 (Pisa, Italy, 25-26 September 2008). Proceedings, pp. 255 - 262. (LNCS, vol. 5247). Springer.

[Peng, 2004] Chengyuan Peng and Petri Vuorimaa. 2004. Text Adaptation for Mobile Digital Teletext. In Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI '04). IEEE Computer Society, Washington, DC, USA, 453-456. DOI=10.1109/WI.2004.133 http://dx.doi.org/10.1109/WI.2004.133

[Perez et al., 1995] Perez, T., J. Gutierrez, and P. Lopisteguy: 1995, An adaptive hypermedia system. AI-ED'95, 7th World Conference on Artificial Intelligence in Education, Washington, DC, pp. 351-358.

[de Rosis, 1996] de Rosis, F., B. De Carolis, and S. Pizzutilo: 1993, User tailored hypermedia explanations. INTERCHI'93 Adjunct proceedings, Amsterdam, pp. 169-170.

[Rekimoto, 1997] Rekimoto, J. (1997) Pick and Drop: A Direct Manipulation Technique for Multiple Computer Environments. In: Proc. of 10th ACM Symposium on User Interface Software Technologies UIST‟97 (Banff, October 14-17, 1997), ACM Press, New York, pp. 31–39.

[Rouillard, 2008] Rouillard, J. 2008. Adaptation en contexte: contribution aux interfaces multimodales et multicanal. Habilitation à Diriger les Recherches (HDR) de l'Université des Sciences et Technologies de Lille1, Laboratoire LIFL, décembre 2008.

[Saloun and Velart, 2006] Petr Saloun and Zdenek Velart. 2006. Adaptive hypermedia as a means for learning programming. In Workshop proceedings of the sixth international conference on Web engineering (ICWE '06). ACM, New York, NY, USA, Article 11 . DOI=10.1145/1149993.1150006 http://doi.acm.org/10.1145/1149993.1150006

[Schlee & Vanderdonckt, 2004] Schlee, M., Vanderdonckt, J., Generative Programming of Graphical User Interfaces, Proc. of 7th Int. Working Conference on Advanced Visual Interfaces AVI‟2004 (Gallipoli, May 25-28, 2004), ACM Press, New York, 2004, pp. 403-406.

[Serna et al., 2010] Audrey Serna, Gaelle Calvary, and Dominique L. Scapin. 2010. How assessing plasticity design choices can improve UI quality: a case study. In Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '10). ACM, New York, NY, USA, 29-34. DOI=10.1145/1822018.1822024 http://doi.acm.org/10.1145/1822018.1822024

[Vanderdonckt et al., 2004] Vanderdonckt, J., Limbourg, Q., Michotte, B., Bouillon, L., Trevisan, D., Florins, M., UsiXML: a User Interface Description Language for Specifying Multimodal User Interfaces, in Proc. of W3C Workshop on Multimodal Interaction WMI'2004 (Sophia Antipolis, 19-20 July 2004).

[Vanderdonckt et al., 2007] Vanderdonckt, J., Coutaz, J., Calvary, G., Stanciulescu, A., Multimodality for Plastic User Interfaces: Models, Methods, and Principles, Chapter 4, in “Multimodal user interfaces: signals and communication technology”, D. Tzovaras (ed.), Lecture Notes in Electrical Engineering, Springer-Verlag, Berlin, 2007, pp. 61-84.

[Vanderdonckt et al., 2005] Vanderdonckt, J., Grolaux, D., Van Roy, P., Limbourg, Q., Macq, B., Michel, B., A Design Space for Context-Sensitive User Interfaces, Proc. of ISCA 14th Int. Conf. on Intelligent and Adaptive Systems and Software Engineering IASSE‟2005 (Toronto, 20-22 July 2005), International Society for Computers and their Applications, Toronto, 2005, pp. 207-214. (FP6 Similar NoE)

[Zeiliger, 1993] Zeiliger, R.: 1993, Adaptive testing: contribution of the SHIVA model. In: D. Leclercq and J. Bruno (eds.): Item banking: Interactive testing and self-assessment. NATO ASI Serie F, Vol. 112, Berlin: Springer-Verlag, pp. 54-65.

Acknowledgements

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

Glossary

- http://serenoa.morfeo-project.org/wiki/index.php/CommonGlossary

Annex A – Adaptation Techniques Templates for Content

Conditional Text

Name:	Conditional Text
References:	[Brusilovsky, 1997]
Description:	All information about a concept is divided into several chunks of text, each chunk is associated with a condition on the level of the user knowledge represented by the user model. When the information is presented to the user by the system only the chunks associated with a pre-defined condition are exhibited.
Rationale:	Given a content, and a user model, a criteria is established and only content information that obey to that criteria is presented according to the user profile.
Example:	A given content may be inappropriate for users under a certain age, so according to her profile the content is adapted to be presented
Context:	According to the context of user, user's profile or preferences.
Advantages:	- Interaction more efficient - Unnecessary content is not exhibited - Better performance
Disadvantages:	The content must be first divided, the criteria well established, and the user model defined
Sample:	
Pictures:	
Observation:	~~T~~his technique is a low-level technique (it requires some "programming" work from the author to set all the required conditions) but it is also very flexible. By choosing appropriate conditions on the knowledge level of the current concept and related concepts represented in the user model the author can implement different methods of adaptation. A simple example is hiding chunks that contain irrelevant explanations if the user's knowledge of the current concept is good enough, or turning on a chunk with comparative explanations if the corresponding related concept is already known. This conditional text technique can be easily applied in the WWW context"

Frame Based

Name:	Frame-based
References:	[Brusilovsky, 1997]
Description:	With this technique all the content about a domain is represented in the form of a frame, slots of frame can contain several different explanations about a content (links, examples). Presentation rules can be used to decid which slots should be presented to which user an in which order. Rules define presentation priorities. To create the rules the user knowledge, or features of the user model can be considered.
Rationale:	The concept is defined, content related is selected, and presented grouped. The criteria to present the content is established.
Example:	Students in an e-learning platform that present difficulties in a specific topic (this can be detect with the analisiys of her exam), can access information about it, during the interaction with the system;
Context:	According to user's preferences, profiles, background, and context of use
Advantages:	The information presented will be adapted to the user level of expertize, improving usability;
Disadvantages:	Pre-processing is necessary, to associate content information, and to define criteria for the presentation of the content
Sample:	
Pictures:	
Observation:	

Stretch

Name:	Stretch
References:	[Brusilovsky, 1997]
Description:	This technique presents to the user different parts of the content according to her knowledge level. With this technique the content in a system can be extended. The idea is to present the content with all extensions if it is relevant for the user, otherwise the extensions of the content are collapsed and not presented
Rationale:	First, the developer define which parts of the content are ~~un~~collapsable", then the remaining of the content, such as additional explanations of a particular concept, or as a low level detail of a particular concept, are linked to the main content. Then a criteria is established in order to define when the content shall be exhibited or collapsed.

Example:	Users with a high level of interest in a domain can access further information about a content of their interest; Or users that have already a certain background information about a domain, do not have to access basic content
Context:	According to the context of use, user‘s preferences or profiles
Advantages:	The usability may be improved
Disadvantages:	First the content has to be produced and the conditions to present or collapse the content need to be defined and related to certain criteria
Sample:	
Pictures:	
Observation:	~~-~~Users with good knowledge of a concept will always find additional explanations of this concept hidden (collapsed) and all low level details uncollapsed. On the contrary, users with poor knowledge of a concept will always find additional explanations of this concept visible and all low level details collapsed. The user with medium level knowledge will see both kinds of information. An important feature of the adaptive stretch technique is that it lets both the user and the system adapt the content of a particular page, taking into account both the knowledge and the preferences of the user. After its initial presentation, the stretched content can be further adapted by the user who can uncollapse and collapse appropriate explanations and details according to her preferences. The system updates the user model according to her preferences to ensure that she will always see the preferred combination of collapsed and uncollapsed parts. For example, if the user has collapsed additional explanations of a particular concept, the system will always show additional explanations of this concept collapsed until the user changes her preferences”

Audio

Audio techniques will be investigated in the future work.

Image

Crop

Name:	Cropping
References:	http://en.wikipedia.org/wiki/Cropping
Description:	This technique cuts an image in order to remove some parts of it
Rationale:	Given an image file, and a criteria to cut, the document is processed and a new version is generated containing only a specific part of its original version
Example:	The systems presents only the faces of people in their avatar pictures
Context:	When some parts of the image aren't necessary for the context or when we want focus on only one part of the image
Advantages:	- Smaller image in size and weight - Faster to load - Better adaptation for mobile devices
Disadvantages:	The result of the use of this technique can lead to hide some parts of the image
Sample:	
Pictures:	

Observation:	

The techniques of: automatic rotation, color, definition, diamond, format, halo, info border, positioning, scale, scrolling, and zooming were already identified. Their templates are under construction and they will be presented in the following reports.

Text

Additional Explanation

Name:	Explanation Variants.
References:	[Brusilovski, 1996], MetaDoc [Boyle & Encarnacion, 1994], KN-AHS [Kobsa et al., 1994], EPIAIM [de Rosis et al., 1993], Anatom-Tutor [Beaumont, 1994], Lisp-Critic [Fischer et al., 1990], Hypadapter [Hohl, Bocker & GunzenhSuser, 1996], ORIMUHS [Encarnacao, 1995], SYPROS [Gonschorek & Herzog, 1995]
Description:	The goal is to hide from the user some parts of information about a particular concept that are not relevant to the user's level of knowledge about this concept. Additional explanations can be given to users who are ready for them. Additional explanations take advantage of users' knowledge to offer more in-depth information to users who can understand it. For example, low level details can be hidden from users a with poor level of knowledge of this concept because they cannot understand these details. On the contrary, additional explanations usually required by novices to understand the concept can be hidden from a user with a good level of knowledge of the concept because they do not need these explanations anymore. In more general terms, in addition to the basic presentation, some category of users can get some additional information, which is specially prepared for this category of users and will not be shown to users of other categories.
Rationale:	The explanation variants method can be implemented by fragment variants and page variants techniques. Page variants is the most simple adaptive presentation technique. With this technique, a system keeps two or more variants of the same page with different presentations of the same content. As a rule, each variant is prepared for one of possible user stereotypes. When presenting a page, a system selects the page variant according to the user profile. These systems store several examples illustrating particular concept and offers the user the example, which is most suitable to the user's previous experience and interests. Fragment variants is a more fine-grained implementation of explanation variants method. For each concept mentioned in the page, the system selects the explanation which is most suitable to the user's knowledge level.
Example:	Students from different levels may require specific definitions in order to comprehend a content. The same text can be presented, but according to the background of the user, she decides whether to see the additional explanation.
Context:	Concerning different users' profiles.
Advantages:	It assumes that showing or hiding some portion of the content is not always sufficient for the adaptation because different users may need essentially different information. With this method, the system stores several variants for some parts of the page content and the user gets the variant which corresponds to his or her user model.
Disadvantages:	It requires extra efforts of the developer in order to: define specific terms that need explanation, create the additional explanation about the term, and link these concepts.

Sample (HTML):	`Textual content`
	`<a id="footnote-1-ref" href="#addexp">[complex term]</a>.`
	`<p id="footnote-1">`
	`- Additional explanation <a href="#addexp">&#8617</a>`
	`</p>`

Pictures:	
Observation:	It is the most popular method of content adaptation; A goal-based variant of this method is to hide from the user some parts of information about a particular concept, which are not relevant to the current user's goal (Hook et al., 1996); The use of anchor is one of the possible approaches to implement this techniques, a link to a new page can also be used though.

Alignment

Name:	Alignment
References:	http://www.w3schools.com/CSS/css_align.asp
Description:	Given a text content, an adaptation process changes its alignment (left, right, centralized or justified). The alignment can consider the vertical or horizontal position of the content.
Rationale:	A text is selected, a alignment type is chosen, the adaptation is performed, the text is presented with the new alignment.

Example:	A user is reading a text, and she prefers a different type of alignment to be applied, then she changes it.
Context:	According to the user preferences or the device type (for instance justifying a text in a large screen may not be a good idea once the user will have more effort to achieve the end of the lines).
Advantages:	The readability may be increased
Disadvantages:	

Sample (HTML, CSS):	`<p align="center">`
	`This text is centered.`
	`</p>`
	`.center`
	`{`
	`margin-left:auto;`
	`margin-right:auto;`
	`width:70%;`
	`background-color:#b0e0e6;`
	`}`

Pictures:	In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, just remember that all the people in this world haven't had the advantages that you've had.'
Observation:	

Altering Fragments

Name:	Altering Fragments
References:	[Brusilovsky, 2002]
Description:	The altering fragments technique can be used to select between a number of alternative explanations, perhaps using different wording to suit different types of users. But the technique can also be used to replace a technical term by a non-technical one for users who have not seen the definition of the term.
Rationale:	Define contexts of use, preparing different versions of the content, present each appropriate version of the content for each context
Example:	A user with a lower level of education/instruction can access an article from a newspaper with a version of the content that better suits for her previous knowledge; Saloun and Velart (2006) describes an adaptive hypermedia system that provides progressively different content (tests) according to the topics that the students learned.
Context:	According to different users' profiles, according to different user's contexts
Advantages:	The content can be comprehended by more people, the accessibility is increased
Disadvantages:	Requires the developers to define different contexts of use as well as build different versions of the content that are more appropriate for each context
Sample:	
Pictures:	
Observation:	This technique is functionally identical to adaptation of modality (according to [Bailey, 2002]). Once fragments can contain multiple media representations of the same data objects. In such cases, choosing the best media type to display (adaptation of modality) is a process of selecting one fragment from a set of fragments (altering fragments).

Background

Name:	Background
References:	www.gmail.com; http://www.w3schools.com/Css/pr_background-color.asp
Description:	This adaptation technique consists in providing different options of background to be changed
Rationale:	Different background options are available, the adaptation technique is performed, the background is modified
Example:	The user wants to choose a new them for accessing her email client, there are options available, she selects one, the background is modified
Context:	According to user preferences, wishes, context of use, or even environment variables (level of light, weather)
Advantages:	User preferences are considered, contrast may be improved making the access to the content more comfortable

Disadvantages:	Not always the user preference is the best choice according to pre-defined guidelines of contrast

Sample (CSS):	`body{background-color:yellow;}`
	`h1{background-color:#00ff00;}`
	`p{background-color:rgb(255,0,255);}`

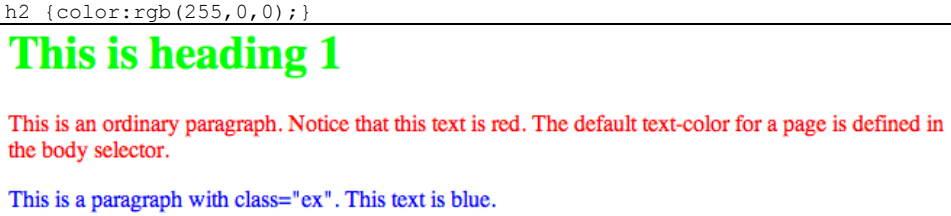
Pictures:	 Different background for gmail, according to user preferences, and also according to the weather of the region of choice (5th row, 1st column)
Observation:	Contrast between background and text color must respect pre-defined guidelines, this requirement provides better readability and accessibility

Color

Name:	Color
References:	http://www.w3schools.com/Css/css_text.asp; http://www.lighthouse.org/accessibility/design/accessible-print-design/making-text-legible
Description:	Change the color of the text
Rationale:	Given a text content, a new color is selected and once the adaptation technique is applied the color of the text is modified accordingly

Example:	The user is writing a code, but the text color is too light, she selects a darker color an the content can be better read
Context:	According to the user profile, preferences, and the level of light of the environment
Advantages:	The new choice may improve the readability, or the user satisfaction
Disadvantages:	Certain combinations of color (regarding background and content) are not appropriate

Sample:	`body {color:blue;}`
	`h1 {color:#00ff00;}`
	`h2 {color:rgb(255,0,0);}`

Pictures:	
Observation:	Users with visual impairments such as color blindness may not perceive colors as able-bodied users, these cases must be considered, as well as a minimum level of contrast that permits a comfortable access to the content; ~~V~~ery high contrasts are difficult to achieve with color combinations other than black and white. Printed material, generally, is most readable in black and white. Different colors may be important for aesthetic or other reasons, but it is better to use such combinations only for larger or highlighted text, such as headlines and titles.”

Comparative Explanation

Name:	Comparative Explanation
References:	
Description:	This technique refers to a comparison between topics described on different pages. The comparison can only be understood by users who have read both pages. So when visiting one of these pages first, the comparison will not be made, but when visiting the other page the comparison appears.
Rationale:	Given two different text contents, an algorithm is applied in order to present to the user the similarities and differences between the contents
Example:	A reader accesses the news from two different sources, with this adaptation techniques the reading may be optimised once only the information that varies between the sources will be accessed
Context:	According to the user‘s preferences, profiles or context of use
Advantages:	The access time will be optimised, interaction more efficient, usability improved
Disadvantages:	Long contents require more effort for processing, and consequently more time is needed to present the result

Sample (PHP):	`<?php`
	`echo strcasecmp("text content 1","Text content 2");`
	`?>`

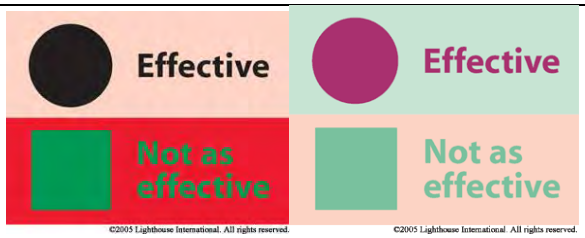
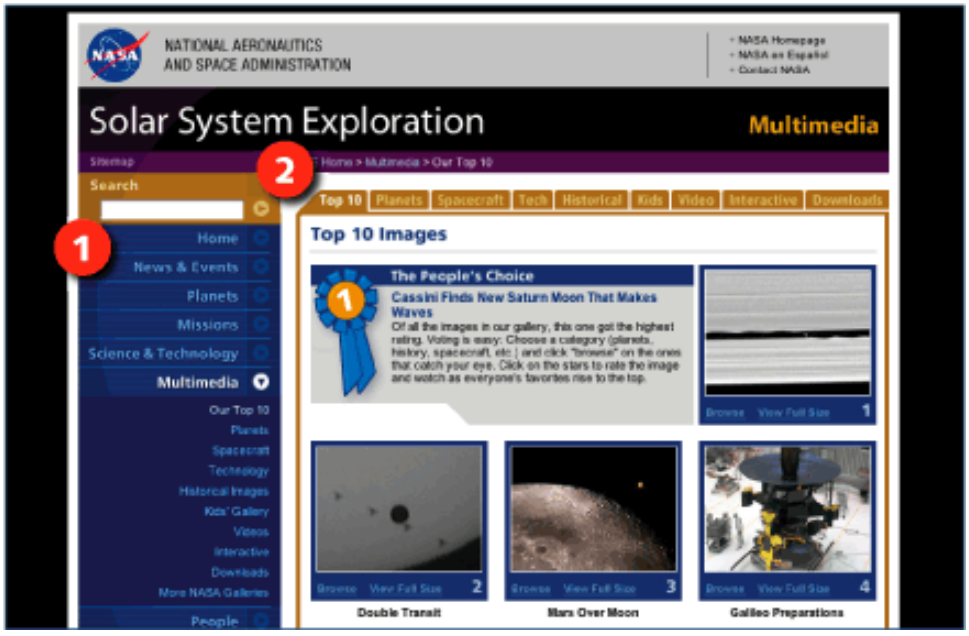
Pictures:	
Observation:	The sample presents a basic function to compare string content, this function only returns whether a difference exists between the two contents (without pointing which it is)

Compare

Name:	Comparison
References:	[Patterno, 1999] Archives & Museum Informatics: MW99 - Papers http://www.archimuse.com/mw99/papers/paterno/paterno.html#ixzz1Ba9BCwLo
Description:	With this method, the system gives the user information that allows her to make the comparison or to provide directly a comparison; there are many types of information that can be analysed to provide comparisons or highlight differences.
Rationale:	Given two text contents, they are compared and the differences between them are pointed
Example:	A system that controls versions of the content can compare it between two different dates and present to the user which were the changes that occurred
Context:	According to the context of use, or user‘s preferences
Advantages:	The user does not have to perform the analysis manually, once the system points the differences between contents
Disadvantages:	Long texts may require a significant effort of processing

Sample:	
Pictures:	
Observation:	It is similar to comparative explanation, the techniques could be possibly grouped

Contrast

Name:	Contrast
References:	http://universalusability.com/access_by_design/text/contrast.html; http://www.lighthouse.org/accessibility/design/accessible-print-design/effective-color-contrast
Description:	It consists in a combination between other two adaptation techniques: change the text color and change the background color, it may increase or decrease the contrast
Rationale:	Given a system interface, its text content and its background color is modified in order to change its contrast
Example:	A user in an airplane, the lights are switched off and the contrast of the book being accessed in an e-reader changes in order to provide a more comfortable reading
Context:	It may adapt according to the level of light in the environment, the device itself, user's preferences, user's impairments (e.g. astigmatism)
Advantages:	The usability of the content will be improved (readability, user satisfaction, more comfortable)
Disadvantages:	Guidelines for a minimum contrast must be taken into account, once user's preferences, or different devices may prevent or difficult the access to the content
Sample:	
Pictures:	  Some of the navigation links on the Solar System Exploration pages may difficult the reading. There is no enough contrast between background and foreground colors. The links in 1 and 2 are particularly difficult to read once they are a lighter shade of the background color.
Observation:	Contrast guidelines must be taken into account (a minimum level of contrast is mandatory in order to permit the reading in different devices and environments), there are online checkers to perform this task –Text should be printed with the highest possible contrast. There is good evidence that for many readers who are older or partially sighted, light (white or light yellow) letters on a dark (black) background are more readable than dark letters on a light background. However, the traditional dark on light may be aesthetically preferable."

Describe

Name:	Describe
References:	
Description:	A description is presented to the user in order to provide further information about a specific content
Rationale:	Given a text content, one word is selected and a description about it is presented to the user
Example:	The user is reading a text content, but she does not understand the meaning of one specific word, the description of it can be accessed
Context:	According to the context of use, user's preferences or profiles
Advantages:	The content can be better understood
Disadvantages:	The developer needs to define specific words, their description, and provide access to it
Sample:	
Pictures:	
Observation:	

Dimming Fragments

Name:	Dimming Fragments
References:	http://wwwis.win.tue.nl/~debra/2ID20/week1/img23.html www.sis.pitt.edu/~peterb/3954-061/Adaptive%20Presentation.ppt
Description:	The text that is not intended for a user is de-emphasized (grayed out, smaller font). With the dimming fragments technique the relevance of a text fragment is not used to hide it, as with stretchtext, but instead to make it less visible. The idea is that the user is stimulated to not read the text, although it is still there and can be read when desired. Shading or graying out can be used to indicate optional reading material. In magazines one often sees so-called ~~-s~~ide bars" with such optional text. It is also possible to imagine a combination of stretchtext with dimming: small pieces of text can be shown like a tooltip, meaning that there is a small icon which when clicked on or when the mouse moves over it shows the text temporarily.
Rationale:	Given a text content, parts of it are dimmed according to pre-defined criteria (such as, content previously accessed are less visible)
Example:	The user is reading a text content, a camera scans her eyes and a tracking algorithm processes her movements, the content previously accessed has its level of transparency increased in order to improve the focus in the content currently read
Context:	According to the context of use, user's preferences or profiles
Advantages:	The reading may be optimised, more efficiency in interaction
Disadvantages:	Users may be not familiar with the technique
Sample:	
Pictures:	
Observation:	This technique can be combined with strecht text to create de-emphasized text that conditionally appears or only appears after some event (like clicking on a tooltip icon).

Explanation Variants

Name:	Explanation Variants.
References:	[Brusilovsky, 1996]
Description:	With this method, the system stores several variants for some parts of the page content and the user gets the variant, which corresponds, to his or her user model.
Rationale:	Prepare different versions of the content according to defined user models, check the user model when she accesses the content, and the present the respective version accordingly
Example:	Students may have access to the content according to their previous background information
Context:	According to the context of use, user's preferences or profiles
Advantages:	The usability is increased (especially user's satisfaction)

Disadvantages:	It requires efforts of the developers to prepare content versions and define user's models
Sample:	
Pictures:	
Observation:	"Explanation variants assumes that showing or hiding some portion of the content is not always sufficient for the adaptation because different users may need essentially different information. With this method, the system stores several variants for some parts of the page content and the user gets the variant which corresponds to his or her user model.—

Font Family

Name:	Font Family
References:	http://www.w3schools.com/css/tryit.asp?filename=trycss_font-family http://www.lighthouse.org/accessibility/design/accessible-print-design/making-text-legible
Description:	The families of the fonts may differ, for instance there are decorative, cursive.
Rationale:	Given a text content, the family of the font may be adapted.
Example:	Text content that are long (such as an article or a chapter of a book) are better read in a standard font with serif, such as roman typefaces.
Context:	According to the content, or the user preferences, or user profile.
Advantages:	The readability may be improved (consequently the usability and accessibility).
Disadvantages:	Some font types may require more space (consequently a scroll bar may be necessary).

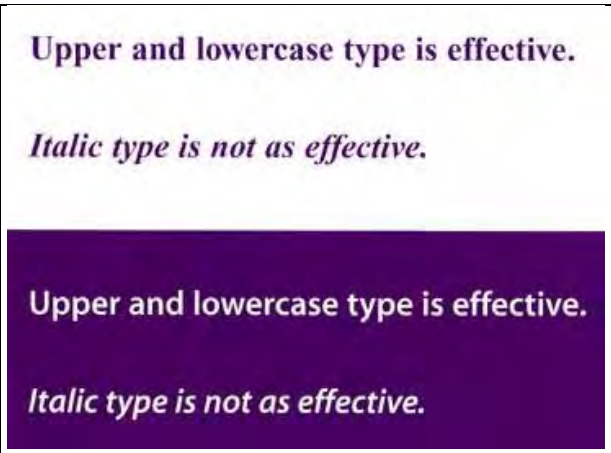
Sample (CSS):	`p`
	`{`
	`font-family:"Times New Roman",Georgia,Serif;`
	`}`

Pictures:	
Observation:	

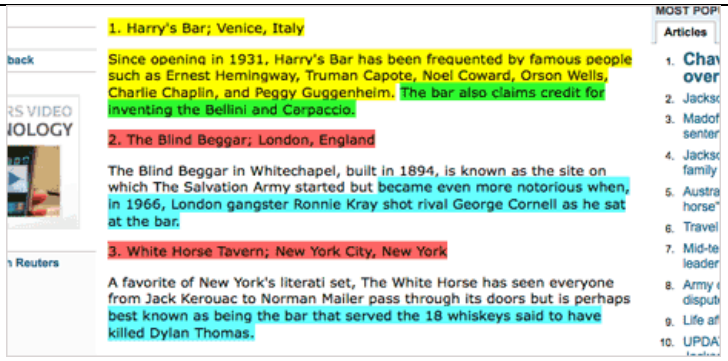
Font Style

Name:	Font Style
References:	http://www.lighthouse.org/accessibility/design/accessible-print-design/making-text-legible
Description:	The style of the font concerns the use of upper and lower cases, as well as italics, oblique or condensed.
Rationale:	Given a text content, its font style is modified.
Example:	In order to emphasize parts of the text, to remark for users that are studying a text content it may be adapted, some words or statements can be underlined, italicized or become bold.
Context:	According to the user's preferences, profile, or context of use.
Advantages:	The text content can be better read or even comprehended.
Disadvantages:	The excesses in style may become the reading less efficient and more uncomfortable (e.g. the use of capital letters)

Sample:	`p.normal {font-style:normal;}`
	`p.italic {font-style:italic;}`
	`p.oblique {font-style:oblique;}`

Pictures:	
Observation:	~~-~~While there is little reliable information on the comparative legibility of typefaces, there is some evidence that a roman typeface, using upper and lower cases, is more readable than italics, oblique or condensed."

Highlight

Name:	Highlight
References:	[Patterno, 1999] Archives & Museum Informatics: MW99 - Papers http://www.archimuse.com/mw99/papers/paterno/paterno.html#ixzz1Ba9BCwLo; http://www.tothepc.com/archives/highlight-selected-text-on-webpages-with-right-click/
Description:	This technique highlights elements that can raise curiosity, discussion or comments from the users;
Rationale:	A text is given as an entry, an algorithm is applied, and a new version with the content highlighted is generated.
Example:	The first text is a long article of a magazine, the new version contains the main ideas highlighted.
Context:	According to the user' preferences or context of use
Advantages:	The content will be remarked, the comprehension of the content may be improved
Disadvantages:	It is not always easy to identify and prioritize the main ideas of the content
Sample:	
Pictures:	 This example illustrates the content highlighted with 4 different colors (it is an add-on of the browser Firefox (Highlights)
Observation:	

Orientation

Name:	Orientation
References:	http://articles.sitepoint.com/article/iphone-development-12-tips/2
Description:	Given a text content its orientation will be modified in angles varying in 360^{o}
Rationale:	Given a text content in a landscape, it will be processed to be presented in portrait mode
Example:	The user is reading a text in a wide screen, however the text respects margins of a A4 paper for printed version, the user prefers to change the orientation of the screen to access the content in an optimized form (more text can be accessed in a given time), the system automatically detects the changing of the orientation of the screen and

	changes the orientation of the text
Context:	User's preferences, profiles, screen's orientation
Advantages:	The reading may be optimized (more space of the screen is used in a given moment), usability and accessibility levels may be improved
Disadvantages:	

Sample:	`function setOrientation() {`
	`var orient = Math.abs(window.orientation) === 90 ? 'landscape' : 'portrait';`
	`var cl = document.body.className;`
	`cl = cl.replace(/portrait|landscape/, orient);`
	`document.body.className = cl;`
	`}`

Pictures:	 Example of different orientation according to the position of the device
Observation:	

Pre-requisite (Explanations)

Name:	Pre-requisite (Explanations)
References:	[Brusilovsky, 1996]
Description:	Before presenting an explanation of a concept the system inserts explanations of all its prerequisite concepts which are not sufficiently known to the user.
Rationale:	First the content is analysed and terms that may require further definitions are defined, The definition of these terms is prepared and becomes available in the interface. A link, or an anchor can be used to provide access to the explanation.
Example:	A page can provide a short introduction or explanation when a unknown term is used (for instance a technical term or a name that the user had no contact before).
Context:	A textual content with terms that may be unknown or that have a difficult level of comprehension for users.
Advantages:	The comprehension of the content is improved.
Disadvantages:	The performance and efficiency to complete a task may decrease (user has to ‗stop' her first goal in order to gain a pre-required knowledge); this deviation may also confuse users (for instance if the explanation is to long or require other pre-requisites, or even if the user has difficulties to follow the course of the interaction).
Sample:	
Pictures:	
Observation:	"The first method is based on prerequisite links between concepts. The idea is the following: before presenting an explanation of a concept the system inserts explanations of all its prerequisite concepts which are not sufficiently known to the user. This method is used in Lisp-Critic [Fischer et al., 1990] and C-book [Kay & Kummerfeld, 1994]." "Prerequisite explanations try to compensate for missing foreknowledge"

Readability

Name:	Readability
References:	
Description:	Given a text content its readability can be increased by applying a set of adaptation techniques
Rationale:	Given a text content: its size may be increased, the spacing may be improved, its alignment may be changed, the colors and contrast may be modified too
Example:	A user with sight impairments in a train, with an iPad, enters in a tunnel while reading an article, the level of contrast of the page will be improved, the size of the font increased and the spacing of the text increased too
Context:	According to the level of light of the environment, available space in the screen, user's profile and preferences
Advantages:	It will become easier and more comfortable for the user to read the text, usability level will be improved, user satisfaction will increase
Disadvantages:	It may change the distribution of the content in the interface
Sample:	
Pictures:	
Observation:	

Serif

Name:	Serif
References:	http://www.lighthouse.org/accessibility/design/accessible-print-design/making-text-legible http://www.w3schools.com/css/css_font.asp
Description:	Serif fonts have small lines at the ends of some characters, such as F
Rationale:	Given a text content its font type can be changed to one with or without serif
Example:	A student is reading a thesis, it is more appropriate to have the content of the chapters in a font with serif and the titles (and short contents) in font without serif
Context:	According to the content, user's preferences, user's profile
Advantages:	The reading may be comfortable, and efficient
Disadvantages:	The size of the content is also affected when the serif is included or removed
Sample:	`p{font-family:"Times New Roman", Times, serif;}`
Pictures:	
Observation:	~~"~~On computer screens, sans-serif fonts are considered easier to read than serif fonts." There are guidelines that recommend serif or not according to the context of use

Similarity

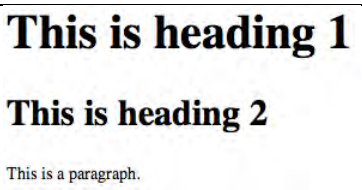
Name:	Similarity.
References:	[Brusilovsky, 1996]
Description:	If a concept similar to the concept being presented is known, the user gets a comparative explanation, which stresses similarities and differences between the current concept and the related one
Rationale:	Given a text content, words, expressions, or sentences are defined, and alternative content is linked to them in order to present related concepts
Example:	The user is reading a text, but she would like to know other definitions for an expression that was found, the system provides similar concepts via a link
Context:	According to the context of use, or user's preferences or profiles

Advantages:	The comprehension of the content will be improved
Disadvantages:	The time to conclude the original task (reading) will be affected
Sample:	
Pictures:	
Observation:	This method is based on similarity links between concepts. If a concept similar to the concept being presented is known, the user gets a comparative explanation, which stresses similarities and differences between the current concept and the related one. Such comparative explanations are particularly effective in the domain of programming languages.

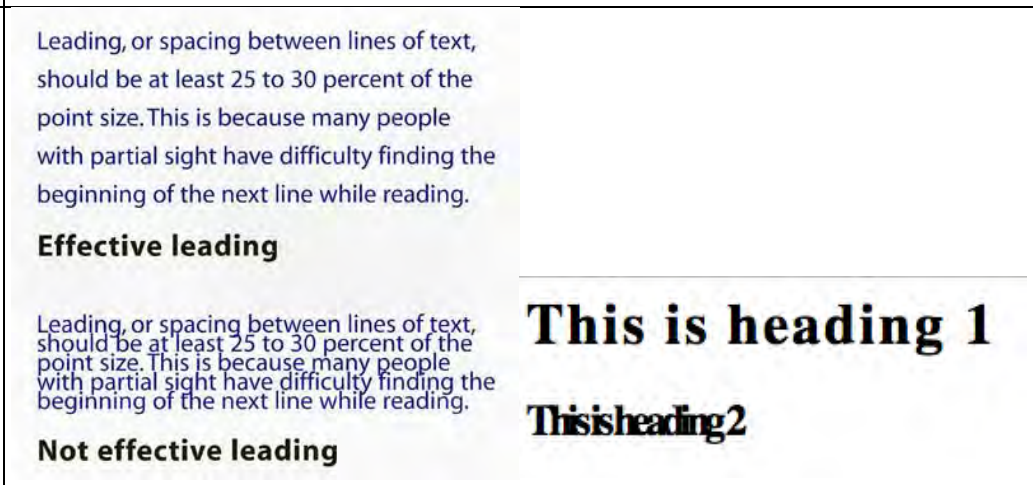
Simplify

Name:	Simplify
References:	
Description:	Given a text content, a new version is produced, based in this original version but aiming a specific type of public
Rationale:	Given a text content, a user's profile, an adaptation technique is applied and a new version of the content is generated and presented to the user
Example:	A technical content may be simplified to be understood also by non-technical readers; children can understand a simplified version of complex contents
Context:	User's profiles, preferences, context of use
Advantages:	Access will be increased, comprehension of the content will be improved
Disadvantages:	It may be a hard task to well-define a target group, its comprehension level, and to implement an algorithm that generates a new version of a text content, that efficiently promotes its understanding
Sample:	
Pictures:	
Observation:	

Size

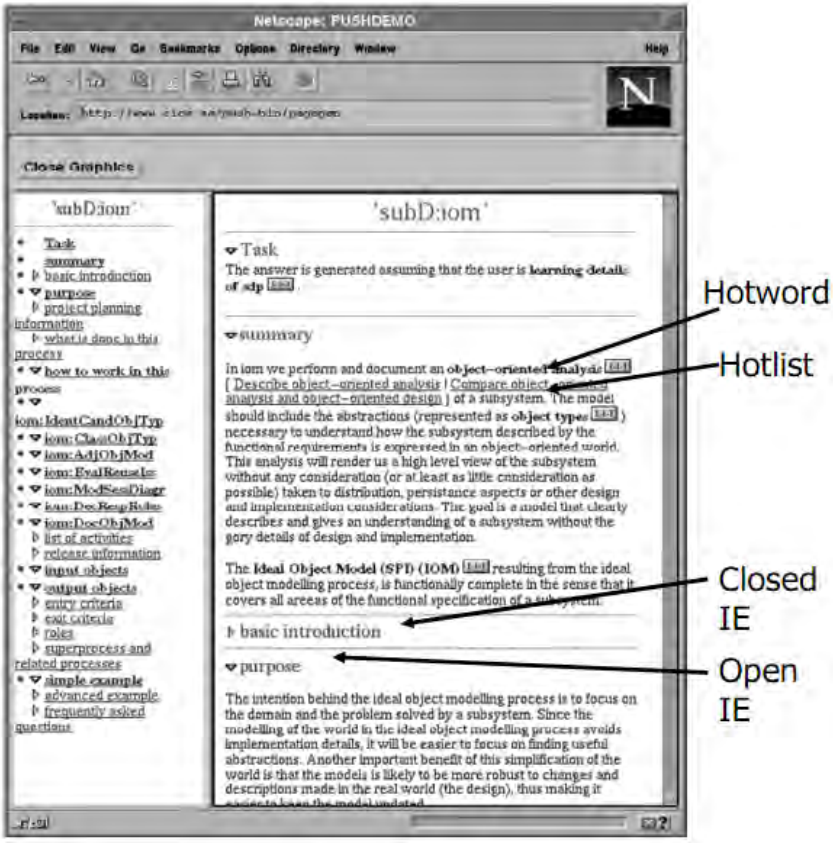
Name:	Font Resize
Reference:	http://www.w3schools.com/css/tryit.asp?filename=trycss_font-size_px; http://www.lighthouse.org/accessibility/design/accessible-print-design/making-text-legible
Description:	Change the font size according to the context.
Rationale:	Given a text content, an adaptation rule is applied in order to change the text size, increasing or decreasing it.
Example:	A visually impaired user accesses a news portal but the font size is inappropriate for reading. The font size can be increased, allowing the user to read the text.
Context:	User with visual impairments, screens far from the user, content in small sizes.
Advantages:	The readability of the text will become possible or it will be improved.
Disadvantages:	The flow of the content may change, parts of the text may be hidden, scrolling may be required, the quality of the content may decrease (according to its resolution).
Sample (CSS)	h1 {font-size:40px;} h2 {font-size:30px;} p {font-size:14px;}
Picture:	
Comments:	The results of the adaptation must be evaluated once the information flow or content distribution may be affected; –Type should be large, preferably at least 16 to 18 points, but keep in mind that the relationship between readability and point size differs somewhat among typefaces." If possible, it is recommended to chose the elements that define content type in HTML (<p>,<h1>,…) instead of manually setting the text size in the code to avoid accessibility and usability issues

Spacing

Name:	Spacing (Leading)
References:	http://www.lighthouse.org/accessibility/design/accessible-print-design/making-text-legible http://www.w3schools.com/css/tryit.asp?filename=trycss_letter-spacing
Description:	Leading is the spacing between lines of text (or between the characters).
Rationale:	Given a text content the spacing between the lines or the characters can be modified in order to provide a better reading.
Example:	An elderly user with a mobile phone reading a text message, according to the user profile the leading may be increased to facilitate the reading.
Context:	User profile, screen size of the device.
Advantages:	Increases the readability of the content, consequently the usability and accessibility of the text.
Disadvantages:	More space in the screen is necessary for the content, a horizontal (or a vertical) scroll bar may be necessary depending on the size of the screen that is available.
Sample:	`h1 {letter-spacing:2px;}` `h2 {letter-spacing:-3px;}`
Pictures:	 Good and bad examples of line and character spacing.
Observation:	

Stretch Text

Name:	Stretch Text
References:	http://wwwis.win.tue.nl/~debra/2ID20/week1/img23.html, http://www.win.tue.nl/~laroyo/2L340/resources/AdaptivePresChapRevised.pdf, [Hook, 1997]
Description:	Items can be open or closed. Adaptive stretchtext is a technique which is essentially an adaptive version of the replacement links in the Guide hypertext system. This system was developed by Peter Brown of the University of Canterbury and was later commercialized by the OWL company. The idea is that items or paragraphs can be displayed or hidden, and that the system decides adaptively which items to open when the page is first displayed. The user can always decide to open or close items at will, and this may give feedback to the system that can be used to change the user model so that the decisions which items to open automatically may change.
Rationale:	In order to give focus to specific parts of the content, less relevant content is defined and hidden
Example:	The user is reading an article, the sections that were already read before are „closed‘ (replaced by an icon that permits access to it)
Context:	According to the context of use, uesr‘s profile or preferences
Advantages:	Only the content of interest is presented and accessed
Disadvantages:	User may not noticed that further information can be also accessed depending on the link type that is used
Sample:	

Pictures:	 Push system exemplifies the technique of Stretchtext
Observation:	

Summarize

Name:	Summarize.
References:	[Patterno, 1999] Archives & Museum Informatics: MW99 - Papers http://www.archimuse.com/mw99/papers/paterno/paterno.html#ixzz1Ba9BCwLo; [Peng, 2004]
Description:	This technique receives as an input a text content and process it in order to obtain a summary of the content. As a result a new version of the text is produced but only the main ideas are included. "after presenting a series of piece of information it can be useful to summarise the main aspects that have been considered; the system can dynamically elaborate some information depending on the data available."
Rationale:	A text is given as an entry, an algorithm is applied, and a new version of the content is generated and presented
Example:	The first text is a long news of a magazine, the new version contains the main ideas as a headline. "if the user examines multiple works of the same author it can be interpreted as a strong interest for such an author and this can trigger the presentation of summarising information on such an author, such as a discussion of the preferred techniques and materials used"
Context:	According to the user's preferences, context of use, or even the size of the screen in the device
Advantages:	The content can be read faster, less space is required for presentation.
Disadvantages:	The information is not completely presented, it is not always easy to identify and prioritize the main ideas of a content.
Sample:	

Pictures:	Digital camera is UK's top gadget A **digital camera** *has been* voted as the **top** gizmo of the **year** *in a* magazine poll of **gadget** *lovers*. Small, affordable **digital cameras** were big this **year**, *but pocket video devices* are expected to be the **gadgets** of 2004. But, he says, *2003* was definitely the **year** of the **digital camera** *with* sales overtaking film **cameras** *for the first* time. Four examples of summary of an article, observe that the outcomes can also have varied lengths
Observation:	

Sort

Name:	Sort
References:	http://www.w3schools.com/xsl/tryxslt.asp?xmlfile=cdcatalog&xsltfile=tryxsl_sort
Description:	Sorting fragments is most useful when a number of more or less independent fragments can be presented in any order. The sorting can be done to perform relevance ranking, like in a list of search results, but it can also be done to show an example before an explanation or the other way around.
Rationale:	Given a text content, and a sort criteria, the text can be ordered accordingly, and presented to the user
Example:	Students in a e-learning platform with different cognitive styles may prefer a different order of descriptions or explanations of a concept, the items in a menu, for instance, can be ordered according to their access rate, which becomes the navigation more efficient; Search results can be ordered according to their access rate;
Context:	Different users have different preferences, and according to her goal it may be interesting to have options of sorting for textual contents; the context of use may also be taken into account (most accessed, popular, requested, new); the user's profile may also define the orders (e.g. blind users may access results according to their accessibility level –metrics can be used in this context-)
Advantages:	The user may find what she is looking for faster, increased efficiency in the interaction
Disadvantages:	The ordering may take a long time to be performed depending on the 'volume' of content to be processed and the ordering criteria chosen

Sample (XSL):	`<xsl:template match="/">`
	`<xsl:for-each select="techniques/adaptation">`
	`<xsl:sort select="domain"/>`
	`<xsl:value-of select="adaptation"/>`
	`</xsl:for-each>`
	`</xsl:template>`

Pictures:	**Adaptation Techniques** Adaptation / Domain Sort / Content Zoom / Content Crop / Content Mapping / Navigation Collapse to Zoom / Presentation Contrast / Presentation Example of content presented sorted by the domain in alphabetical order.
Observation:	

Translate

Name:	Translate
References:	www.google.be

Description:	Given a text content in a specific language, an adaptation technique is applied and the text is translated to another language
Rationale:	A text is selected, its language is known, another language is specified, the text is translated and then presented
Example:	A user travels to a country with a different language and she can access the news of this country in her own mother tongue
Context:	According to the user's preferences, profiles, region, context of use
Advantages:	The text can be comprehended, accessibility is improved
Disadvantages:	According to the size of the content the processing can take a long time, some algorithms were implemented to provide translation, however not always the results are correct
Sample:	
Pictures:	 In this example the browser Chrome recognizes the country of origin of the request (Belgium), the preferences of the user signed in, and asks the user whether she wants to have the page translated
Observation:	Providing different versions of the content can also be a solution (instead of automatically processing the content), however it takes a significant effort of the developers; In this domain there are two possibilities: translation and location, according to the consideration of the ‚cultural' aspects of the region where the language is spoken; Providing versions to different languages helps to achieve the universal access

Truncate

Name:	Truncate
References:	http://www.w3schools.com/XPath/xpath_functions.asp
Description:	Consists in given a text content, its size is reduced by simply cutting part of it, the text can be cut respecting paragraphs, words, or not
Rationale:	Given a text with n characters (or words or sentences), the content after a given point is removed, the content supressed can be replace with a sign, such as „...", or with a link that gives access to the complete content, further criteria can be taken into account (such as respecting words, or sentences)
Example:	In a search engine, the user usually access the title of the page, and part of the content which was previously truncated; It can be also applied for privacy reasons, for instance when only the first 4 characters of a telephone number is presented;
Context:	In small screens, according to user's preferences, or context of use (when space is limited or insufficient)
Advantages:	The user can have an idea about a variety of information without having to access the complete content, space in the interface is saved, more results can be presented, and only information of interest accessed in more details
Disadvantages:	It is hard to establish the amount of text content that is enough for the user to understand if that information is interesting for her; Truncating content in the middle of a word or sentence may not be a good approach (e.g. the content may have a wrong meaning and interpretation)

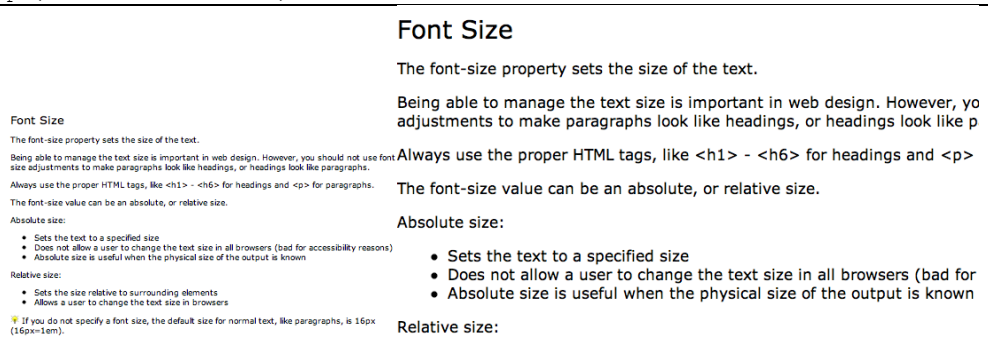
Sample (XSLT):	`fn:substring(string,start,len): returns the substring from the start position to the specified length. Index of the first character is 1. If length is omitted it returns the substring from the start position to the end`
	`Example: substring('97015995',1,4)`
	`Result: '9701'`

Pictures:	 Example of a Google result truncated, the remaining content is replaced by a “...”
Observation:	

Zoom

Name:	Zoom
References:	http://www.w3schools.com/css/css_font.asp
Description:	It consists in modifying the size of the font, increasing or decreasing it
Rationale:	Given a text content, and a command to change its size to become bigger or smaller, the size is modified and the content is presented
Example:	A user with visual impairment may prefer to read a text in bigger size
Context:	User‘s profile, user‘s preferences, context of use, size of the screen, position of the screen (according to how far it is from the reader)
Advantages:	The content can be accessed in a more comfortable way
Disadvantages:xslt	The distribution of the content will vary (a new distribution may be required)

Sample (CSS):	`body {font-size:100%;}`
	`h1 {font-size:2.5em;}`
	`h2 {font-size:1.875em;}`
	`p {font-size:0.875em;}`

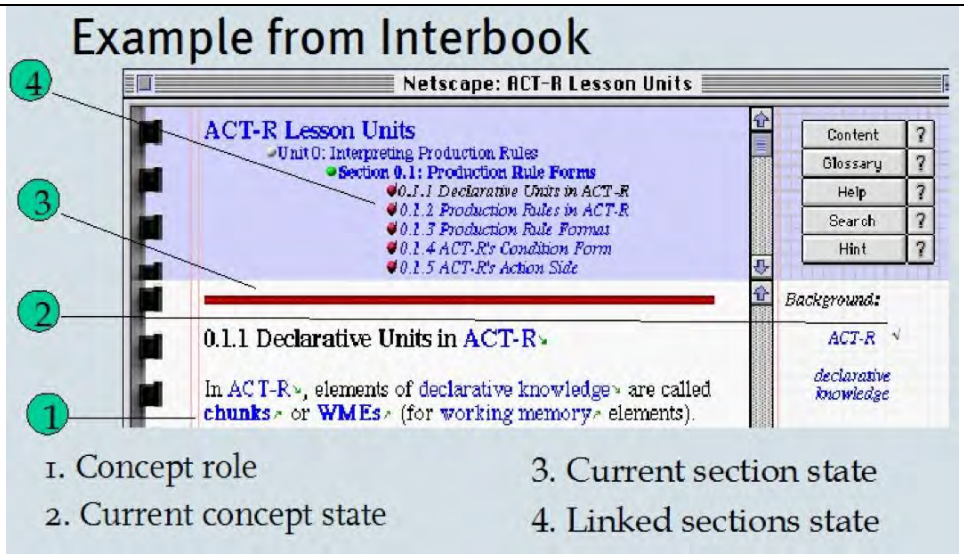
Pictures:	 Example of zoom out and zoom in in the browser
Observation:	Combining the use of *em* and *percent* is an approach that works for all browsers when the user changes the zoom. Setting the font size with *em* avoid a problem with Internet Explorer (the resizing of the whole content instead of only the text), many developers use *em* instead of *pixels*. The *em* size unit is recommended by the W3C. 1*em* is equal to the current font size. The default text size in browsers is 16*px*. So, the default size of 1*em* is 16*px*. The size can be calculated from pixels to *em* using this formula: pixels/16=*em*

Video

Video techniques will be investigated in future works.

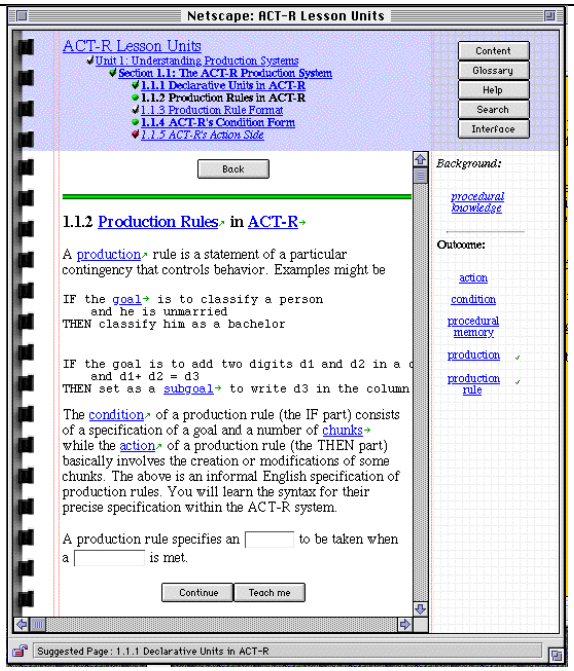
Annex B – Adaptation Techniques Templates for Navigation

Annotation of Links

Name:	Annotation
References:	[Brusilovsky, 1996]; http://wwwis.win.tue.nl/~debra/2ID20/week1/img25.html; AHA!; Interbook, ELM-ART; Inspire; KBS Hyperbook
Description:	All links are visible but an annotation indicates their relevancy. The link anchors may be changed (e.g. in color) or additional annotation symbols can be used.
Rationale:	Given a set of links, and a criteria to annotate them (e.g. popularity, recommended), the adaptation process adds to the links some form of note, by changing the color, adding an icon, etc
Example:	A user has already read some parts of the content before, the system marks these contents as read, by adding a check icon before the link
Context:	According to the context of use, or user's preferences, profiles
Advantages:	It avoids problems with incorrect mental maps. It is powerful once it distinguishes several levels of relevancy for 'interaction'.
Disadvantages:	It may overload the user
Sample:	
Pictures:	 This example shows adaptive link annotation in Interbook. Links in the partial table of contents are annotated with colored balls and checkmarks. Green means the link is recommended, red means not recommended, white means that the page contains no new knowledge. The checkmarks indicate how much the user already knows about the concepts of the page. No checkmark means no knowledge, a small checkmark some knowledge and a large checkmark full knowledge. Note that in the example there is something strange because the links to subsections 0.1.1 through 0.1.5 are not recommended even though the user already has a lot of knowledge about them. The recommendation however depends on the knowledge of prerequisites, not of the concepts treated in the pages themselves. In the page the links are annotated with arrows. These indicate whether the link goes down or up in the hierarchical structure of the course. They are not adaptive. The only adaptive part in the page is the red bar. It indicates that the page the user is reading is not recommended.
Observation:	"Annotation seems to be a very relevant form of adaptive navigation support. Annotation can be naturally used with all four possible forms of links. This technique supports stable order of links and avoids problems with incorrect mental maps. Annotation is generally a more powerful technology than hiding: hiding can distinguish only two states for the nodes while annotation, as mentioned above, up to six states, in particular, several levels of relevancy as it implemented in Hypadapter [Hohl, Bocker & GunzenhSuser, 1996]. Annotations do not restrict cognitive overload as much as hiding does, but the hiding technology can be quite well simulated by the annotation technology using a kind of "dimming" instead of hiding for "not relevant" links. Dimming can decrease cognitive overload in some extent (the user can learn to ignore dimmed links), but dimmed links are still visible (and traversable, if required) which protects the user from forming wrong mental maps." -Adaptive link annotation is the most popular link adaptation technique. It is the least restrictive technique: all the links are accessible. Annotations are used to indicate how interesting the link is for the user. Many systems use some kind of icon in front of or behind the link anchor to indicate the relevance of the link. Since the Web has been extended with style sheets it has also become possible to use the color of the link anchor as an annotation. This is

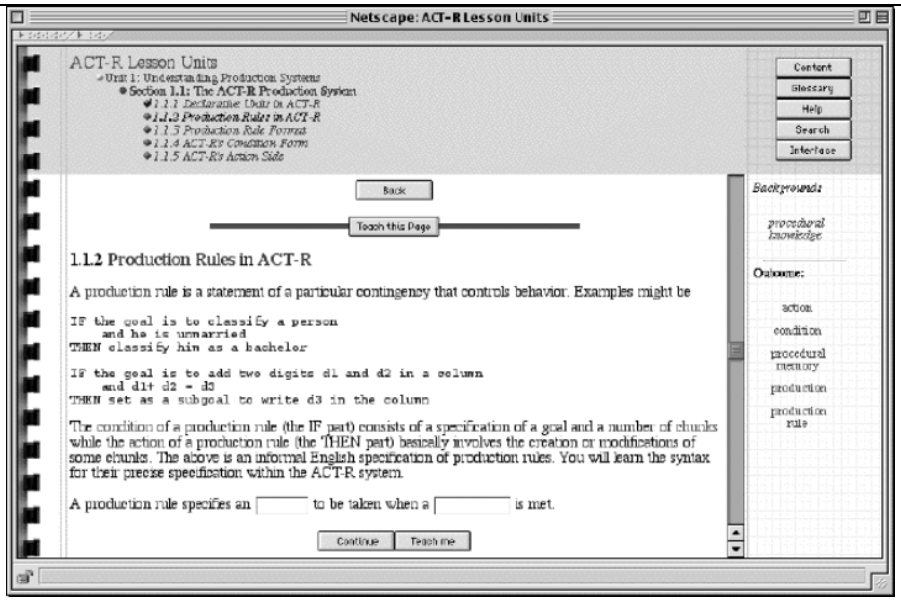
	not without drawbacks: some users are so used to links on the Web being blue or purple that they do not recognize words in other colors as being link anchors."

Direct Guidance

Name:	Guidance
References:	[Brusilovsky, 1996]; [Armstrong et al., 1995]; [Brusilovsky et al., 1998];
Description:	This technique decides and presents to the user the next best option for interaction, according to the user's goal and other parameters being considered.
Rationale:	Given a pre-defined criteria, the system suggests to the user the next steps that she should follow (criteria can take into account previous interactions recorded in log file, for instance)
Example:	A student is accessing an e-learning platform, after she accessed, read and studied a content the system provides a link to an example of exam, the user then can try to solve the exercises
Context:	According to the context of use, or user's preferences, profiles
Advantages:	User interaction may be optimised
Disadvantages:	Users can be frustrated in case the suggested links are not interesting for them
Sample:	
Pictures:	 In ACT-R, the Teach me button (bottom right) provides direct guidance for the users.
Observation:	Direct guidance is the simplest technology of adaptive navigation support. Direct guidance can be applied in any system, which can decide what is the next "best" node for the user to visit according user's goal and other parameters represented in the user model. To provide direct guidance, the system can outline visually the link to the "best" node as it is done in Web Watcher [Armstrong et al., 1995], or present an additional dynamic link (usually called "next") which is connected to the "best" node as in ISIS-Tutor [Brusilovsky & Pesin, 1994], SHIVA [Zeiliger, 1993], HyperTutor [Perez et al., 1995], and Land Use Tutor [Kushniruk & Wang, 1994]. The former way is more clear, while the latter is more flexible, because it can be used to recommend the node which is not connected directly to the current one (and not represented on the current page). Direct guidance is a clear and easy to implement technology, it can be used with all four kinds of link presentation listed above. The problem with direct guidance is that it provides limited support: "follow me or no help". Direct guidance can hardly be the primary form of navigation support because it provides no support for the users who would not like to follow the system's suggestion. Direct guidance is useful but it has to be used together with a "more supportive" technology.

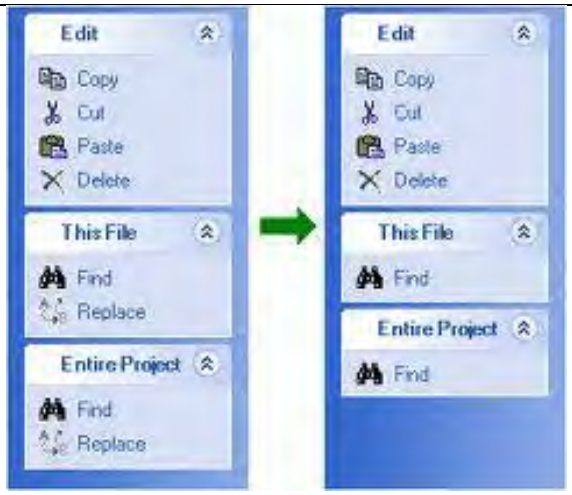
Generation of Links

Name:	Adaptive Link Generation
References:	[Brusilovsky, 2003]; http://wwwis.win.tue.nl/~debra/2ID20/week1/img24.html
Description:	The system may discover new useful links between pages and add them, the system may use previous navigation or page similarity to add links, and generating a list of links is typical in information retrieval and filtering systems.

Rationale:	Given two (or more) different contents in a systems, links are generated between them according to defined criteria (such as an analysis of the previous interactions of the user)
Example:	Students rate different aspects of the system and compare these ratings with the students of different levels of knowledge of the subject. Knowledge Sea applies a neural network-based mechanism to process a large number of pages from different Web-based tutorials along with a set of closed corpus documents (such as lecture notes) and group then by similarity. As a result, a user with a specific educational goal — such as to do readings associated with a particular lecture — can use an automatically generated list of relevant links to explore
Context:	User profile, preferences or context of use
Advantages:	Links of interest may be generated improving the efficiency of the navigation
Disadvantages:	It is necessary an inference algorithm to automatically and efficiently generate the links
Sample:	
Pictures:	 This example illustrates link generation in Knowledge Sea System
Observation:	

Hiding of Links

Name:	Hiding (Link Removing; Link disabling)
References:	[Brusilovsky, 1996]; http://wwwis.win.tue.nl/~debra/2ID20/week1/img25.html; AHA!; http://documentation.devexpress.com/#WindowsForms/CustomDocument4887
Description:	This technique restricts the navigation space by hiding links to "not relevant" pages. Pure hiding means the link anchor is shown as normal text (the user cannot see it as a link). Link disabling means that the link does not work; it may or may not still be shown as if there is a link. Link removal means the link anchor is removed (and consequently it cannot be used). A combination of hiding and disabling is also possible. It indicates that the link anchor text is just plain text. Basically, in link hiding the navigation is simplified and restricting the navigational space supports the user‘s orientation.
Rationale:	Given a set of links in a system, a criteria is established to define which ones are relevant or not, according to the context, links that are considered as not relevant in that context of use are hidden
Example:	A student can access a text content, but according to her previous knowledge, links to basic concepts may be hidden, optimising the navigation
Context:	Context of use, user‘s profile, user‘s preferences
Advantages:	It reduces the complexity of navigation for the users, and it also reduces their cognitive overload. It has a wide applicability; it can be used with all kinds of non-contextual, index, and map links and with contextual links. It is also more transparent to the user and more stable
Disadvantages:	Hiding can distinguish only two states for the nodes - relevant and non relevant; in case of ‗wrong‘ inference users may be prevented of accessing specific content
Sample:	`<style type="text/css" media="all">`

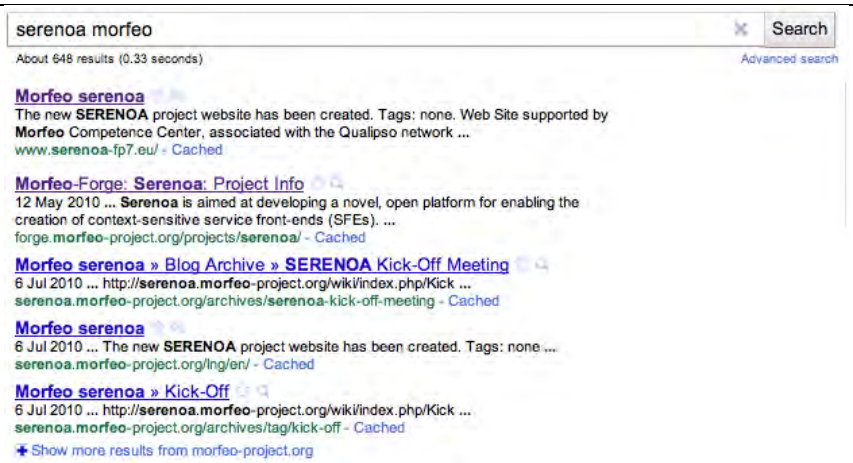
	<!-- .hiddenlink { color: #000; /* same color as the surrounding text */ text-decoration: none; /* to remove the underline */ cursor: text; /* to make the cursor stay as a text cursor, not the hand */ --> </style>
Pictures:	 Example illustrating how links (replace) can be hidden in a menu
Observation:	~~A~~daptive link hiding means that links that are not considered relevant for this user are hidden, disabled or removed in some way. Link hiding means that the link anchor cannot be seen as being a link anchor. When the text on a page is black, a black link anchor, not underlined, looks just like plain text. If the link is still there many browsers will show a special cursor when the mouse pointer is moved over the anchor. The link can also be disabled, meaning that the anchor text is no longer a link anchor. On the Web this is easy to realize by removing the anchor tag. However, that performs hiding as well as disabling. It is possible to use font color and optionally underlining to make the anchor still look like a link anchor, but this is seldom done because it is frustrating for users to see link anchors that do not work as links. Link removal means that the anchor text is removed, thereby automatically disabling the link as well. Link removal can easily be done in a list of links, but not in running text because the user needs to be able to read the text. When asked in an informal setting a large majority of users has indicated that they preferred links in a list to be annotated or ~~h~~idden", but not removed."

Map Adaptation

Name:	Map Adaptation
References:	http://wwwis.win.tue.nl/~debra/2ID20/week1/img27.html
Description:	The map adaptation is a form of adaptive navigation support. In order to give users an idea of the whole hyperspace, and some orientation support regarding where the user is in this space, many applications offer some kind of map. Websites often offer a textual sitemap, mostly because this is easy to generate. A graphical map, preferably based on conceptual relationships rather than link relationships, is a better tool for giving insight into the application's structure. Basically, map adaptation concerns adapting graphical representations of the global and local hyperspace link structure using any of the techniques mentioned in this domain or their combination.
Rationale:	Given the navigational structure of the systems, its links are graphically represented, such as a map, in order to present to the user possible interaction routes
Example:	The user in an ATM wants to transfer money to an international account, however the first interface does not provide an option to do so, she can then look for and access the map of the system to find the route for performing her task
Context:	According to the context of use, or user's profile and preferences
Advantages:	The navigation may be improved, users can perform their tasks more efficiently
Disadvantages:	Large maps are not insightful; a map can adaptively be reduced so that the user can still grasp the overall picture. Nodes on the map can also be annotated to indicate relevance, to indicate where the user has gone before, and perhaps even to indicate where other users have gone.
Sample:	

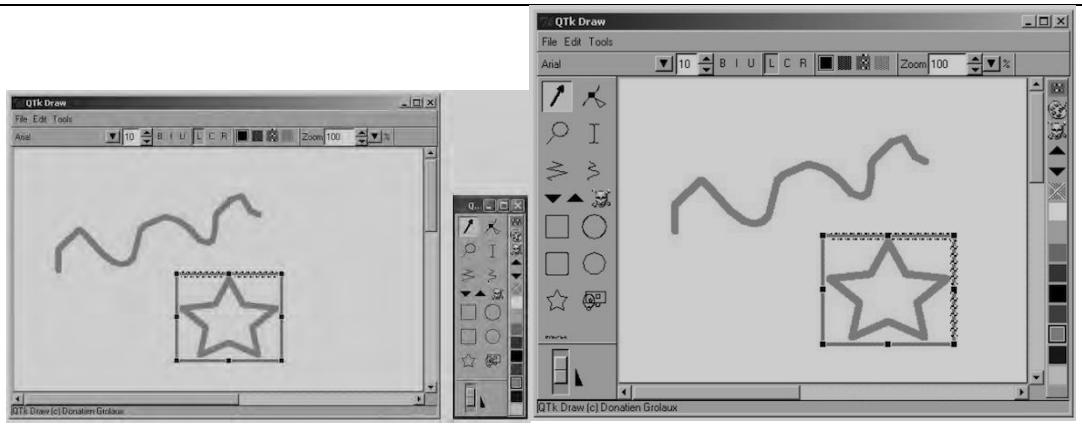
Pictures:	
Observation:	Complete site maps are not feasible for non-trivial hyperspace; a local or global map can be adapted by annotating or removing nodes or larger parts; a map can also be adapted by moving nodes around; maps can be graphical or textual; adaptation can be based on relevance, but also on group presence.

Sorting of Links

Name:	Sorting of Links
References:	[Brusilovsky, 1996]
Description:	This technique sorts all the links according to the user and user-valuable criteria: the closer to the top, the more relevant the link is.
Rationale:	Given a set of link options for the user to access, they are ordered according to a pre-defined criteria and then presented to the user, criteria can be: most popular, most accessed, alphabetical order
Example:	The student in an e-learning platform has recommended contents to read, the teacher attributes to the documents priorities in reading them, the system presents the links to the documents based in the priority order for the student to read them
Context:	According to the context of use, user's preferences, user's profiles
Advantages:	Useful for information retrieval. It can significantly reduce the navigation time. It can improve the usability level (user's satisfaction, user's efficiency in performing a task)
Disadvantages:	Limited applicability: it can be used with non-contextual links, but it can hardly be used for indexes and content pages and it can never be used with contextual links and maps. Technology can make the order of the links non-stable, it may change each time the user enters the page
Sample:	
Pictures:	 Google provides the search results and their links according to a specific algorithm that considers different variables, such as meta-information of the websites
Observation:	

Annex C – Adaptation Techniques Templates for Presentation

Attach

Name:	Attach
References:	[Groulax et al., 2005]
Description:	Attach is a property in which part of an interface can be attached to the another interface being used so as to recompose another one on-demand, according to user's needs, task requirements. "Any UI component of interest can be attached back to its previously detached UI or to any other UI. Thanks to the attachability property, it is possible to support a UI development process by copy/paste. In traditional visual programming, any UI is drawn by composition of widgets dragged from a tool palette onto a working area. This process does not support per se composition of new UI from previously defined UIs. Of course, it is possible to copy/paste parts of the widgets, but there is a need to redraw everything. In Programming by demonstration, a UI that will be implemented is demonstrated and then derived. When a UI component is attached to another UI component, they are automatically merged so as to create an entirely new UI. There is no need to redraw the UI and this operation can be done at run-time."
Rationale:	Given an interface, a new one (entirely or partly) is merged to it. "Any selected component from one UI can be copied, dragged and dropped into another UI to compose a new UI merging functions which are the sum of functions provided by the individual components."
Example:	In a calendar the user wants to join also a widget with weather information.
Context:	When there is a need or a wish to have more functionality available at the same interface.
Advantages:	Improving the functionality of the application, adapting to specific needs or wishes.
Disadvantages:	It may be not easy to implement, it may overload the space, affecting performance for instance. It may require redistribution.
Sample:	
Pictures:	 In this example the colour pallet was attached to the QTk Draw application.
Observation:	In scenarios with situations where the user may want to compose, decompose and re-compose the components of a UI on-demand, depending on users' needs, task requirements and platforms availability, an interaction paradigm, called Detachable User Interfaces is considered. It is characterized by the „Demi-Plat', a set of properties that includes Attachability. Attachability is the property in which the plastified UI component is attached to any UI running on the target-computing platform, if needed.

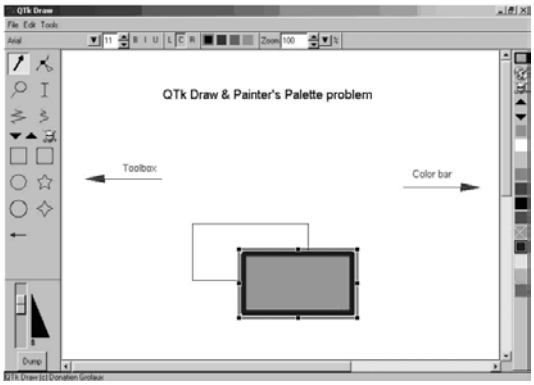
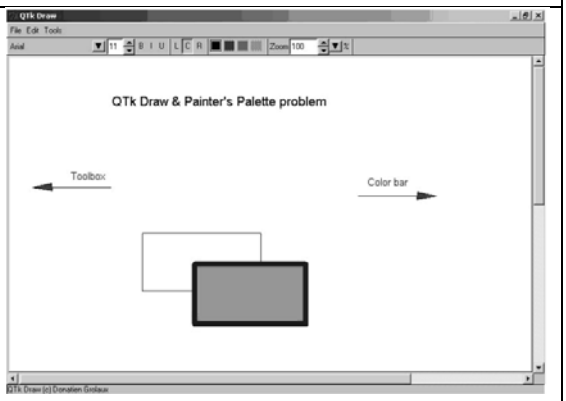
Collapse to Zoom

Name:	Collapse to Zoom
References:	[Baudisch et al., 2004]; [J. Coutaz, 2007]
Description:	Collapse-to-zoom, which allows users to collapse areas of web pages deemed irrelevant,
Rationale:	Given an interface, composed of interactors, the original position of the resources is modified, according to pre-defined criteria. In addition to allowing users to zoom into relevant areas, collapse-to-zoom allows users to collapse areas deemed irrelevant, such as columns containing menus, archive material, or advertising. Collapsing content causes all remaining content to expand in size causing it to reveal more detail, which increases the user's chance of identifying relevant content. Collapse-to-zoom navigation is based on a hybrid between a marquee selection tool and a marking menu, called marquee menu. It offers four commands for collapsing content areas at different granularities and to switch to a full-size reading view of what is left of the page. Collapse-to-zoom uses space gained from

	collapsing tiles to immediately magnify the remaining page content.
Example:	(a) As the page loads, the browser detects that the page is significantly bigger than the screen and thus shows a thumbnail view instead. (b) The headlines are scaled below recognisability, so the user cannot decide if any of the shown headlines involve the topic of interest. However, layout conventions allow the user to recognize the main structure of a news page, i.e., menu, articles, and advertising columns. To see the news headlines in more detail, the user performs a pen gesture to collapse the advertising column and then (c) the menu column. (d) As a result, these columns are now replaced with thin gray place-holders and the freed screen space has been used to render the remaining column with the news headlines at a larger size. While the abstracts are still fairly small, headlines have now become readable. Note how the placeholders provide the user with visual context. In addition, they allow the user to restore the respective tile. Tapping the left placeholder, for example, would restore the view to the state. In the expectation to return to this news site, the user bookmarks it. (e) The user scrolls through the page and finds a headline containing the name Bertelsmann. Without leaving overview mode, the user follows the link by tapping it. (f) The article page is loaded and since it is too long to fit into the browser window, it too is displayed as an overview. To be able to read the story the user performs an expand-and-zoom gesture on the story area. (g) This removes all content outside the story area and switches to single-column view. The user can now advance through the story efficiently using the device‘s hardware scrolling buttons. (h) The next day, the user invokes the bookmark created earlier and the browser loads the new edition of the page. However, the browser also restores the collapse-state the page was in when the bookmark was created. All headlines are therefore readable right away; no further user interaction is required.
Context:	Overview visualizations for small-screen web browsers were designed to provide users with visual context and to allow them to rapidly zoom in on tiles of relevant content. Given that content in the overview is reduced, however, users are often unable to tell which tiles hold the relevant material, which can force them to adopt a time-consuming hunt-and-peck strategy.
Advantages:	Collapsing content causes all of the remaining contents to be redrawn in more detail, which increases the user‘s chances of identifying relevant content. When finally switching to the full-size view, the page has been reduced significantly, which allows users to scroll through the remaining content in an efficient way. The collapse-to-zoom mechanism described in this walkthrough minimizes the need for zooming and trial-and-error exploration. Instead of forcing users to iterate between overview and detail view, collapse-to-zoom allows users to make continuous progress towards the page content that is relevant to the user‘s task at hand. By collapsing elements deemed irrelevant users narrow down the page, thereby reducing the amount of content that has to be examined when finally switching to detail view. The need to scroll over irrelevant content is eliminated. In addition, collapsing content causes the remaining material to be re-rendered with increased detail, which often enables users to continue zeroing in on the relevant material. In addition, the page loads faster, as image material in the collapsed areas is not loaded.
Disadvantages:	Users not always are aware of a scrolling need, what can prevent she to access the content
Sample:	
Pictures:	

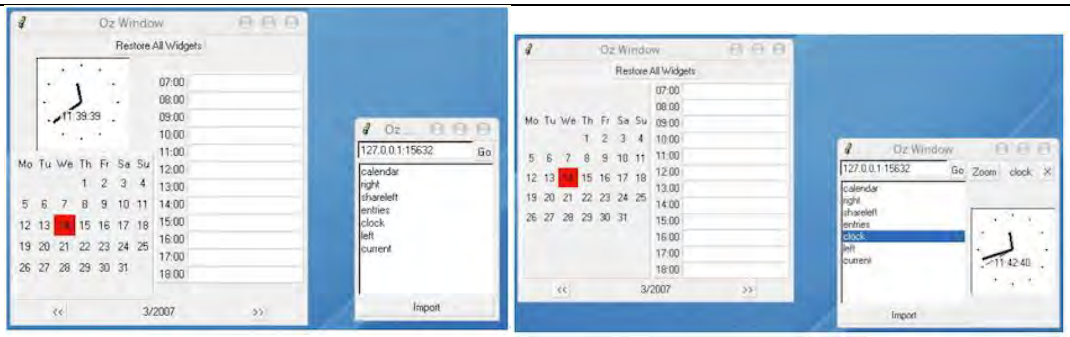
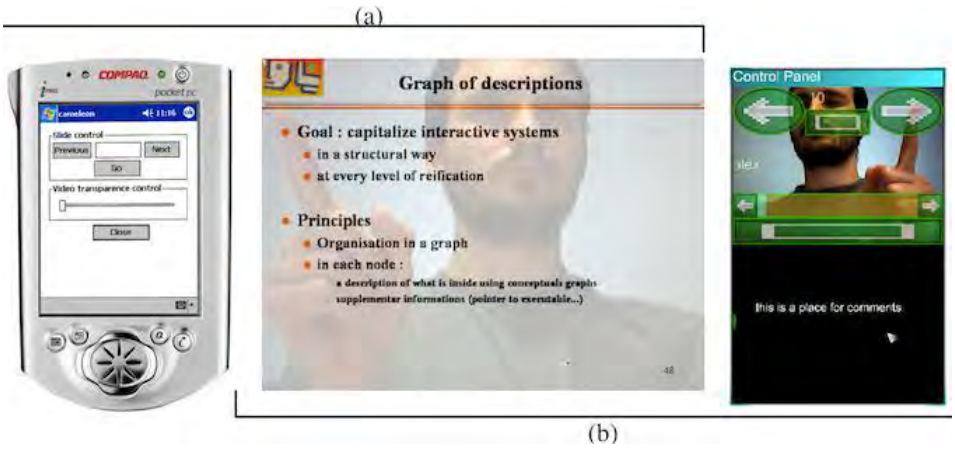
Observation:	With collapse-to-zoom, all page navigation is done up front (often using only a single expand-and-zoom action). Once users switch to single-column view, subsequent navigation reduces itself to scrolling, which most devices support conveniently with a set of hardware scroll buttons. All the expand/collapse navigation is based on drag gestures allows collapse-to-zoom to use pen tapping for following links

Detach

Name:	Detach
References:	[Groulax et al., 2005]
Description:	Detach is a property in which a part of an existing interface can be removed from the current interface. This part can be used to recompose another interface on-demand, according to user's needs and task requirements for instance. Detachable user interfaces consist of graphical user interfaces whose parts or whole can be detached at run-time from their host, migrated onto another computing platform while carrying out the task, possibly adapted to the new platform and attached to the target platform in a peer-to-peer fashion. De-taching is the property of splitting a part of a UI for transferring it onto another platform.
Rationale:	Given an interface, part(s) of it can be removed.
Example:	In an agenda application the user wants to remove the events on top and use it just as a calendar.
Context:	When there is a need or a wish to have fewer resources available at the same interface. It can happen for memory, processing, connection or performance constraints, for instance. Adaptation may consider the availability of platforms and contex of use (tasks)
Advantages:	Making the resources of an application more flexible and adaptable to specific needs or wishes.
Disadvantages:	It may be not easy to implement, once the 'part' of the interface and the dettaching process need to be well-defined
Sample:	
Pictures:	  In this example the toolbox and the color bar were both detached from the original interface of QTk Draw.
Observation:	Different types of detachability: 1. Full screen when the entire UIs of all applications running on the current platform are detached. 2. Window when an entire user/system-selected window or any portion of it is detached. For instance, a whole window within the border, along with its title bar, its menu bar, the scroll bar or captions lines. 3. Active window when the windows that has the focus of interaction on the desktop is detached when the detach operation is invoked. 4. Region when any user-defined rectangular region of the UI is detached. For in- stance, a user may select by direct manipulation a rectangle surrounding components subject to detachment. 5. Fixed region when a user-defined rectangular fixed region of the platform desktop defined by absolute pixel coordinates. 6. Widget when any individual widget is detached.

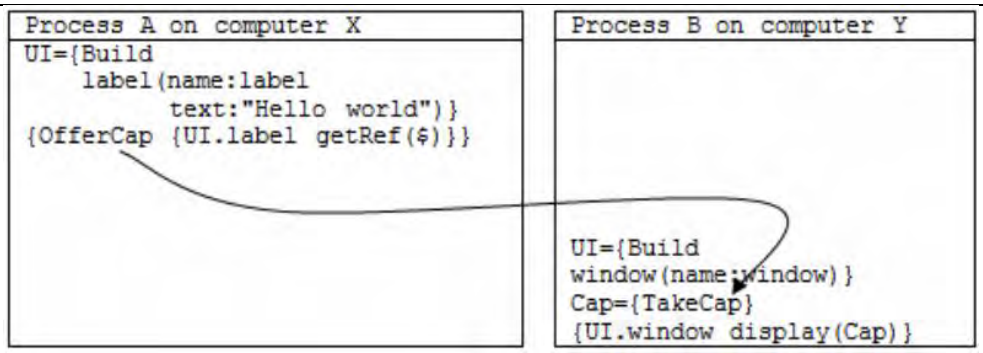
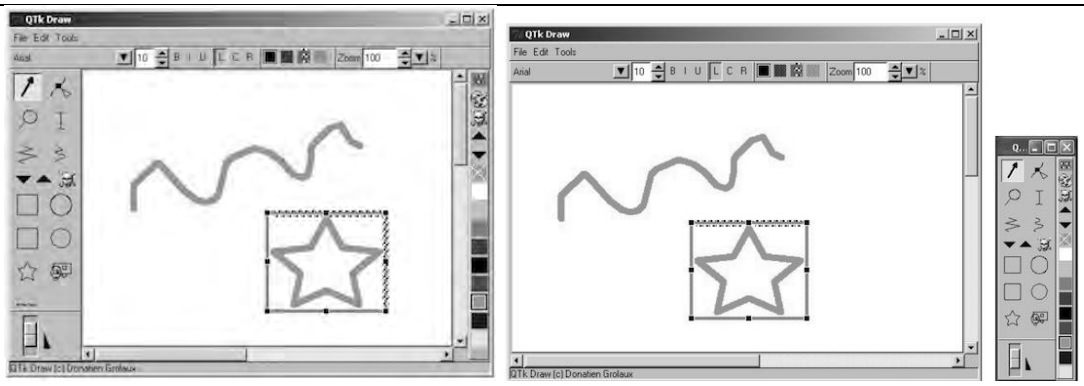
Distribution

Name:	Distribution
References:	[Balme et al., 2004]; [Melchior et al., 2009]
Description:	A Distributed User Interface (DUI) consists of a UI having the ability to distribute parts or whole of its components across multiple monitors, devices, platforms, displays, and/ or users.
Rationale:	The application is composed by resources that may be distributed, once there is the possibility to distribute this resources, the context is considered and adaptation rules are applied, the final version of the application contains resources distributed in different domains (such as: platforms, devices or users)
Example:	In the second picture there is a Pebbles-like configuration where the graphical UI is distributed across the surfaces of a PC and of a PDA. The slides viewer is displayed in a rotative canvas so that it can be oriented appropriately when projected on a horizontal surface (e.g., a table). If the PDA disappears from the cluster, the control panel automatically migrates to the PC screen. Because different resources are now available, the panel is plastified. As

	shown in Figure b, the retargeted control panel includes different interactors, but also a miniature representation of the speaker's video is now available. During the migration-retargeting process, users can see the control panel emerging progressively from the slides viewer while rotating so that they can evaluate the state of the transition. The UI, which was distributed on a heterogeneous cluster is now centralized on an elementary platform. The new UI results from a dynamic partial migration and retargeting at the workspace level. Conversely, if the PDA re-enters the platform, the UI automatically switches to the configuration of Figure a and the control panel disappears from the PC screen by weaving itself into the slides viewer before reappearing on the PDA. [Balme et al., 2004]
Context:	Adapt to different devices, platforms and according to the user preferences.
Advantages:	More comfortable to visualize the content, flexibility of access to the system, user's satisfaction is increased
Disadvantages:	The implementation must be carefully done to separate concerns properly, and to provide users an efficient transition and interaction options to adapt.
Sample:	
Pictures:	 In this example the clock was distributed and can be accessed in a different application. 
Observation:	Distribution can be performed between different: monitors, devices, platforms, displays and users.

Migration

Name:	Migration
References:	[Groulax et al., 2005]; [Patterno et al., 2008]
Description:	Migration consists of transferring any UI component (presentation and dialogue states) from one platform to another.
Rationale:	The migration can be of the entire applications (which is problematic with limited-resource devices and different CPU architectures or operating systems), or the migration can consider only a user interface part of a software application. One approach for migration can be the encapsulation of all volatile execution state of a virtual machine; (however this approach mainly supports migration of applications among desktop or laptop systems by making copy of the application with the current state in a virtual machine and then copy the virtual machine in the target device, which does not address the support of different interaction platforms supporting different interaction resources and modalities, with the consequent ability to adapt to them. An specific approach considers migration of applications shared among several users: when migration is triggered the environment starts a fresh copy of the application process in the target system, and replays the saved sequence of input events to the copy in order to ensure that the process will get the state where it left off. This solution does not consider migration across platform supporting different interaction resources and modalities and consequently does not support run-time generation of a new version of the user interface for a different platform. Another approach assumes that the desktop version of an application exists, without posing any restriction on the method or tool used for its development. Then, during the user session, a version for the platform is dynamically generated exploiting model-based techniques [Paterno

	et al., 2008].
Example:	A user who starts interacting with a digital museum application providing information about artworks. At some point the user accesses a page, which allows him to specify preferences regarding artworks in the museum. While the various preference options are specified, the application using an Ajax script provides a preview of artworks that satisfy them without requiring access to a new page. Then, the user asks for migration to a mobile device in order to continue by means of indicating preferences on the move.
Context:	The available platforms may be considered in order to perform this adaptation technique
Advantages:	They allow users to move about freely, change device and still continue the interaction from the point where they left off
Disadvantages:	It may be not easy to implement: to obtain usable migration, preserving the user interface state across multiple devices and adapting to the changing interaction resources
Sample:	 Migration between different platforms
Pictures:	 This example shows a colour palette that was detached from QTk Draw, it migrated from a TabletPC to a Desktop.
Observation:	Migration can be characterized along four axes, according to the amount of: - Platforms: the migration can be one-to-one (from one platform to another one) or one-to-many (from one platform to many platforms); - Users: the migration is said to be single-user, respectively multi-user, when it occurs across platforms owned by one user, respectively by many users; - Platform types: the migration is said to be one-threaded, respectively multi-threaded, when it occurs between platforms of the same type (e.g. between two PCs), respectively of different types (e.g., from a PC to a PDA that does not necessarily run the same operating system); - Interaction surfaces: the migration can be mono-surface, respectively multi-surface, when it occurs from one interaction surface to another (e.g., from screen to screen), respectively from one surface to multiple surfaces at the same time (e.g., from one screen to several different screens of various sizes) In migration, the devices change but not the user, who still has the same interests and may appreciate the possibility to find in the current bookmarks including the pages that were bookmarked in the previous device. Another element that has similar characteristics is the browser home page: in some cases users may be interested to migrate it to different platforms as well.

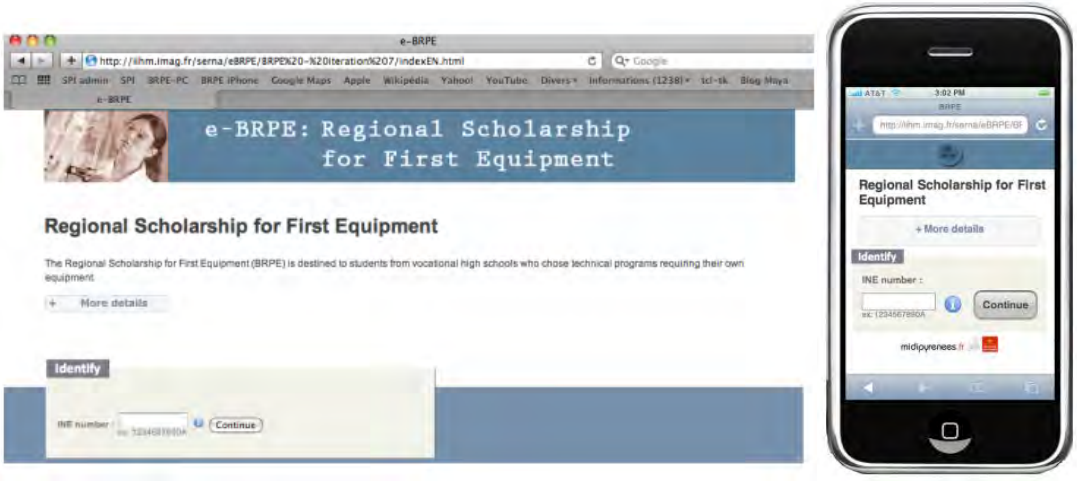
Re-distribution

Name:	Re-distribution
References:	[J. Coutaz, 2007]
Description:	UI re-distribution, i.e. the application of objects re-distribution to UI components, denotes the re-allocation of the UI components of the interactive space to different interaction resources.
Rationale:	Given an interface, composed of interactors, the original position of the resources is modified, according to pre-

	defined criteria.
Example:	If there are 3 columns of content, like text, picture and a table, and the new context involved a mobile phone instead of a large screen, the resources instead of being positioned side-by-side, they are presented in a vertical sequence.
Context:	When a new display is used, the resources may be re-distributed in order to provide a better visualization of them.
Advantages:	It provides a more comfortable visualization of and interaction with the resources. Ease of finding resources. Effectiveness and time-saving.
Disadvantages:	It may be not easy to define the best approach to re-organize the distribution of the resources in an interface.
Sample:	
Pictures:	 In this example (Sedan-Bouillon website) the content may be centralized to be presented on a PC or distributed between PC and a PDA.
Observation:	

Re-molding

Name:	Re-molding
References:	[J. Coutaz, 2007]; [Serna et al., 2010]
Description:	Object re-moulding is to reshape objects without distorting their role. Applied to user interface components, UI re-moulding denotes the reconfiguration of the user interface that is perceivable to the user and that results from transformations applied to the source user interface. UI transformations include: suppression of the UI components that become irrelevant in the new context of use; insertion of new UI components to provide access to new services relevant in the new context of use, reorganization of UI components by revisiting their spatial temporal dependency. Reorganization may result layout and/or their insertion of UI components. Re-moulding may result in using different modalities, or in from the suppression and/or exploiting multimodality differently.
Rationale:	Given an interface, composed of resources, such as interactors, they can be inserted or supressed, according to pre-defined rules, in order to accommodate better in a new context.
Example:	From a large screen displaying a text content to a small screen in a mobile phone, it may be better to present the content in audio format.
Context:	It may be adapted according to the availability of new platforms and devices: when a new display is used, the resources may be re-molded in order to provide a better visualization of and interaction with them.
Advantages:	It provides a more comfortable visualization of and interaction with the resources. Effectiveness and time-saving.
Disadvantages:	It may be not easy to define the best approach to re-define the modalities of the resources in an interface.
Sample:	

Pictures:	 In this example the content is remolded to be better accommodated in a different device
Observation:	Re-molding may occur at different abstraction levels: ranging from the task level to the concrete presentation