

# Designing Web-based Services

N. Partarakis<sup>1</sup>, C. Doulgeraki<sup>1</sup>, M. Antona<sup>1</sup>, C. Stephanidis<sup>1,2</sup>

<sup>1</sup> Foundation for Research and Technology – Hellas (FORTH)  
Institute of Computer Science  
Heraklion, Crete GR-70013, Greece  
cs@ics.forth.gr

<sup>2</sup> University of Crete, Department of Computer Science

**Abstract:** This Chapter presents the state of the art regarding techniques, methods and tools for designing web-based services. Several design techniques, such as user centered design and iterative prototyping, as well as accessibility and usability guidelines, are overviewed and discussed. This analysis leads to the consideration that, in the context of web design, it is critical to provide design techniques and tools that can cope with a diverse user population in terms of social, cultural and personal characteristics. Towards addressing this need, the present Chapter proposes a Web User Interfaces (WUIs) development toolkit, named EAGER, which embodies consolidated design knowledge and supports web designers in addressing diversity through WUIs adaptation. The interfaces created are able to adapt to the interaction modalities, metaphors and user interface elements most appropriate to individual user and context profiles. The proposed design approach is analyzed in depth together with examples of design outcomes.

## 1. Introduction

Recently, the web has become associated with a great amount of daily user activities, such as work, communication, education, entertainment, etc., which are usually supported through web services. A web-based service typically includes a number of facilities for access and navigation of information, socialization, collaboration and trade. It can be defined as a website acting as a gateway to access a multitude of online destinations that are somehow related and which can be referred to as the contents of the service. For example: (a) web pages, websites and other web-based applications (including other web-based services); (b) people connected on the Web<sup>1</sup>; and (c) digital resources<sup>2</sup>, such as documents, multimedia, software, etc. Typically, web-based services also include various reusable web components that display relevant information to portal users such as news, email, weather information, and discussion forums. As a consequence of the opportunities offered by web services, the target population of interactive technologies has widened dramatically with respect to previous generations. Users are no longer only the traditional able-bodied, skilled and computer-literate professionals. Instead, users are potentially anybody. Additionally, a wide variety of mobile devices, which make the web available on the move, are emerging. As a result of this evolution, modern web user interfaces (WUIs) face the challenge to cope with the increasing volume and variance of user needs and requirements, necessitating increased accessibility, usability, personalization, and device independence.

However, despite the universality of the Web and the predominant role of web services in the Information Society, current approaches to web design are not suited to taking into account diversity. Therefore, the design of WUIs, which meet the needs and requirements of as many users as possible, is evolving into an increasingly difficult and demanding task for Web designers. A vast majority of web-designers today compromise on designing a Web application for the “typical” or “average” user, taking this as the best solution to cater the needs of the broadest possible population. Unfortunately, this approach leads to the risk of excluding various categories of users, such as non-expert IT users, the very young and elderly, people with disability, etc. [26]. Specialized designs for one user group often constrain potential use by other

---

<sup>1</sup> In this case the term *community portal* is used.

<sup>2</sup> In this case the term *digital library portal* is used.

important groups. Instead, design needs to be equally targeted towards all potential users. Admittedly, this is a difficult endeavor in the web environment.

Contemporary users desire and expect the delivery of interfaces that ensure personalization and satisfying levels of freedom to decide which way to interact with the Web. In this context, the concept of automatic adaptation is gradually emerging as a means to personalize content and interaction on the web. Recently, few Web applications started providing some sort of adaptation to their users. Indicatively, the *iGoogle web site*<sup>3</sup> and the *Microsoft Sharepoint portal server*<sup>4</sup> offer users the ability to customize the user interface (UI), such as repositioning, minimizing and maximizing webpage elements, defining the number of search results to be displayed, and personalizing the color settings. Clearly, although some of these features are well appreciated by some users, they cannot alone support the delivery of *qualitative user experience for all*, regardless of the user's (dis)ability, skills, preferences, and context of use. For instance, the particular ways in which most of these features are implemented render the produced webpages completely inaccessible for blind individuals that use screen readers.

Consequently, today the main challenge for WUI designers and researchers is to come-up with approaches that can meet diverse user needs and requirements in various contexts. In this Chapter, generic design methods and tools widely used today for achieving the aforementioned goals are discussed, leading to the conclusion that, although powerful, these approaches taken in isolation usually lead to monolithic designs instead of flexible and personalized design. However, recent research efforts have provided important outcomes regarding design methods and techniques (e.g., User Centered Design and Iterative Prototyping), accessibility and usability guidelines, novel general design and development approaches targeted to address diversity, and specialized design support tools.

This accumulated knowledge can be exploited to address current shortcomings in the context of novel development approaches aiming to cope with the diversity of the target user population and the context of use of modern web based applications. In these terms, the main goal of this Chapter is to contribute to the collective vision to mainstream and radically improve the accessibility, usability and, in general, user experience and acceptance of computer-based products and services, and thereby also

---

<sup>3</sup> <http://www.google.com/ig>

<sup>4</sup> <http://sharepoint.microsoft.com/sharepoint/>

help in reducing the part of the population currently not using Information and Communications Technologies (ICT).

In particular, following a discussion of existing knowledge, techniques and tools, proposes a novel approach, based on UI adaptation techniques, to embed personalized accessibility features deep into Web design. Such an approach is incarnated in a WUI development toolkit, named EAGER.

## **2. Design methods and tools**

This section overviews prevalent, as well as more recent design methods, guidelines, development methodologies and tools, and outlines how various approaches and methods have been integrated and embedded in the development of a toolkit that supports WUIs adaptation, thus facilitating web designers in creating WUIs, which can respond to the increasing need for optimal accessibility, usability and device independence.

### **2.1 User centered design**

User centered design [7] is an approach to interactive system design and development that focuses specifically on making systems usable. It is an iterative process whose goal is the development of usable systems, achieved through the involvement of potential users during the design of the system.

User centered design includes four iterative design activities, all involving direct user participation, as shown in Figure 1:

1. understand and specify the context of use, the nature of the users, their goals and tasks, and the environment in which the product will be used;
2. specify the user and organizational requirements in terms of effectiveness, efficiency and satisfaction; and the allocation of function between users and the system;
3. produce designs and prototypes of plausible solutions; and
4. carry out user-based assessment.

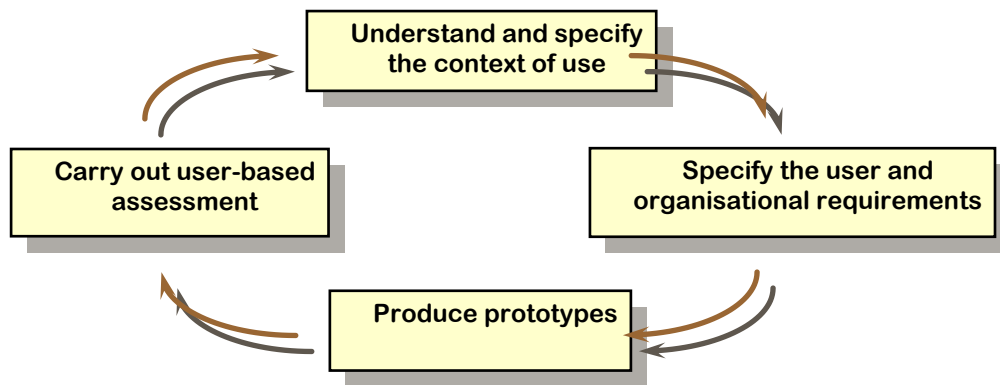


Figure 1. The activities involved in user centered design (from [30])

User centered design requires [7]:

- *Active involvement of users and clear understanding of user and task requirements.* The active involvement of end-users is one of the key strengths, as it conveys to designers the context of use in which the system will be used, potentially enhancing the acceptance of the final outcome.
- *The appropriate allocation of functions between the user and the system.* It is important to determine which aspects of a job or a task will be handled by users and which can be handled by the system itself. This division of labor should be based on an appreciation of human capabilities, their limitations and a thorough grasp of the particular demands of the task.
- *Iteration of design solutions.* Iterative design entails receiving feedback from end-users following their use of early design solutions. The users attempt to accomplish “real world” tasks using the prototype (see section 2.2). The feedback from this exercise is used to further develop the design.
- *Multi-disciplinary design teams.* User centered system development is a collaborative process which benefits from the active involvement of various parties, each of whom have insights and expertise to share. Therefore, the development team should be made up of experts with technical skills in various phases of the design life cycle. The team might thus include managers, usability specialists, end-users, software engineers, graphic designers, interaction designers, training and support staff and task experts.

User centered design provides human-centered protocols and tight design-evaluation feedback loop replacing techno-centric practices with a focus on the human aspects of technology use. However, it is limited in its possibility to address the diversity of user

requirements, as it fosters the traditional perspective of “typical” users interacting with a desktop machine in a business environment [25]. While user centered design focuses on maintaining a multidisciplinary and user-involving perspective into systems development, it does not specify how designers can cope with radically different user groups whose requirements are not known a priori. In particular, with the advent of the web and the emergence of a highly distributed and collaborative computing paradigm, it is difficult for designers to anticipate who the user may be.

## **2.2 Iterative prototyping**

In the context of user-centered design, iterative prototyping can be considered as the process of receiving feedback by end users for facilitating the iterative design of a system. Usually, an iterative user interface design process initiates with the production of low-fidelity prototypes and continues with higher-fidelity prototypes. The use of prototypes in the design phase aims at allowing the designers to test some emerging ideas for the design in question. While evaluating a prototype, the designers can identify functional requirements, usability problems and performance issues that can be dealt with at once and before the implementation phase [21]. The analytical process is conducted through a number of iterations. Each of them involves the creation of prototypes and the evaluation of these prototypes by usability experts and system end-users. A prototype is an easily changeable draft or simulation of at least a part of an interface [6].

Facade tools allow the creator to specify input behavior next to the drawings and text, something which is not possible with pencil and paper. These prototypes, which look and feel like the actual application, operate on a limited set of artificial data but nonetheless effectively show to users the impact of their actions [2]. Explanatory prototypes are drawings of prospective layouts of the system. They are usually very detailed - concerning typography, color schemes, navigation and graphic elements. The tool most commonly used to produce such prototypes is Microsoft PowerPoint. PowerPoint is widely known, and users are familiar with it, while changes can be done quickly with higher precision than if drawn by hand. Power Point fulfils the requirements for good prototyping postulated by [27]: “Ease of use, Fast turn around, Flexibility, Useful throughout the development cycle, Executable and Version control” [10]. The outcome of this phase, the prototype, allows designers and developers to work through the details of the system without doing extensive, time

consuming, and expensive design and programming. Prototyping, in general, assists users to examine and alter things while the design is still flexible, before the final system is implemented. Categories of prototyping tools are summarized in Table 1.

Table 1. Categories of prototyping tools

	low	intermediate	high
<b>Fidelity</b>			→
	Mock-ups	Facade tools	Rapid application development
<b>Examples</b>	Paper & Pencil Sticky notes Plastic objects Whiteboard	Tcl/Tk Hypercard Power point	Delphi Visual basic NeXT UIMX for X C++ CAD

Prototyping has wide application in design and is a very good way of bringing users in contact with applications and services under development in an iterative involvement scheme. However, it should be considered that prototyping alternative design solutions for different needs and requirements using prevalent prototyping tools may become a complex and difficult task if the number of alternatives to be produced is large and no specific support is provided for structuring and managing the design space.

### 2.3 Accessibility Guidelines

Guidelines play a key role in the adoption of web accessibility and usability by industries and society. In essence, they constitute a rapidly evolving medium for transferring established and de facto knowledge (know-how) to various interested parties.

Concerning accessibility, a number of guidelines collections have been developed (e.g., [29], [19]). In particular, the Web Content Accessibility Guidelines (WCAG) [32] explains how to make Web content accessible to people with disabilities. Web "content" generally refers to the information in a Web page or Web application, including text, images, forms, sounds, etc. WCAG 1.0, published in 1999, provides 14 guidelines that are general principles of accessible design. Each guideline has one or more checkpoints that explain how the guideline applies in a specific area. WCAG foresees 3 levels of compliance, A, AA and AAA. Each level requires a stricter set of conformance guidelines, such as different versions of HTML (Transitional vs. Strict)

and other techniques that need to be incorporated into code before accomplishing validation. Further to WCAG 1.0, in December 2008, the W3C announced a new standard that will help Web designers and developers to create sites that better meet the needs of users with disabilities and older users. Drawing on extensive experience and community feedback, WCAG 2.0 [33] improves upon WCAG 1.0 and applies to more advanced technologies.

Another source of web accessibility guidance comes from the US government Section 508 of the US Rehabilitation Act [28], a comprehensive set of rules designed to help web designers make their sites accessible. A website has also been developed where web developers can take online training course for free to learn about these rules.

In general, for a website to comply with accessibility standards, it should have at least the following characteristics:

- (X)HTML Validation from the W3C for the pages content
- CSS Validation from the W3C for the pages layout
- At least WAI-AA (preferably AAA) compliance with the WAI's WCAG
- Compliance with all guidelines from Section 508 of the US Rehabilitation Act
- Access keys built into the HTML
- Semantic Web Markup
- A high contrast version of the site for individuals with low vision
- Alternative media for any multimedia used on the site (video, flash, audio, etc)

The usage of guidelines is today the most widely adopted process by web authors for creating accessible web content. This approach has proven valuable for bridging a number of barriers faced today by people with disabilities.

Unfortunately, however, many limitations arise due to a number of reasons. These include the difficulty in interpreting and applying guidelines, which require extensive training. Additionally, the process of using, or testing conformance to, widely accepted accessibility guidelines is complex and time consuming. To address this issue, several tools have been developed enabling the semi automatic checking of html documents. Such tools make easier the development of accessible web content especially due to the fact that the checking of conformance does not rely solely on the expertise of developers. Developers with limited experience in web accessibility can use such tools for evaluating web content and without the need to go through a large number of check – lists [31].



As a final consideration, guidelines provide a ‘one-size-fits-all’ approach to accessibility, which, while ensuring a basic level of accessibility for users with various types of disabilities, does not support personalization and improved interaction experience.

## **2.4 Usability Guidelines**

Web usability is concerned with how easy or difficult web-based software is to learn and use [35]. Usability is typically defined to include ease of use, ease of learning, efficiency, memorability, user satisfaction and reduction of errors. Today, a considerable body of knowledge dedicated to the usability of web content are available, usually codified in usability guidelines. Guidelines can be found in many different formats with contents varying both in quality and level of detail, ranging from ill-structured common sense statements to formalized rules ready for automatic guidelines checking. Among other sources (or an overview see [8]), seminal books on web accessibility were published, such as [10] and [12]. Specific collections of guidelines for the web address, for example, issues such as screen resolution and page layout [14], the usability of homepages [12], the usability of navigation [13] and of site maps [16], the usability of email [3], the usability of search facilities [11], and the user experience in ecommerce [15]. Guidelines are also available for age groups, such as older users (e.g., [17]), children (e.g., [18]), and teenagers. Usability of web interfaces on mobile devices is also another very timely issue [34]. Finally, an important aspect related to the user experience in web portals and services is sociability [21].

Although web usability guidelines provide a very rich source of design guidance, their use is not straightforward [8]. First, usability remains a quality factor of user interfaces that is still handled with uncertainty. Applying guidelines is a necessary condition, but not a sufficient one: the respect of guidelines certainly contributes to improve the usability of a web site, but a web site that is compliant with all possible guidelines may still be experienced as unusable by some end users. Second, identifying the guidelines appropriate to address a particular web site for a given target audience remains challenging. Little or no guidance exists to provide assistance to developers to locate, select, and gather guidelines relevant to their web site. Finally, guidelines are often not usable by themselves. Often guidelines are not precise enough to be applied unambiguously and to be assessed objectively once applied. By

necessity, different guidelines are often in conflict, as they address different issues and indeed, different requirements. Hence, the prioritization of guidelines becomes necessary [20]. However, usability guidelines alone cannot lead to the development of WUIs adequately responding to diversity.

## **2.5 Design for All**

The usage of the techniques discussed in the previous sections leads to the generation of software that is more reliable user friendly, and therefore more acceptable. However, these techniques, although useful, fail to address the need of providing seamless access to all regardless of the specific characteristics of the user or the context of use. In this context, the term Design for All denotes an effort to unfold and reveal the challenges of accessibility and usability, as well as to provide insights and instrument appropriate solutions in the Human-Computer Interaction (HCI) field [26]. The fundamental vision is to offer an approach for developing computational environments that cater for the broadest possible range of human abilities, skills, requirements and preferences.

Design for All in the Information Society is the conscious and systematic effort to proactively apply principles and methods, and employ appropriate tools, in order to develop IT & T products and services which are accessible and usable by all citizens thus avoiding the need for posteriori adaptations, or specialized design. Design for All in HCI recognizes, respects, values and attempts to accommodate the broadest possible range of human abilities, requirements and preferences, eliminates the need for ‘special features’ and fosters individualization and end-user acceptability.

Design for All fosters a proactive strategy, postulating that accessibility and quality of interaction need to be embedded into a product at design time. This entails a purposeful effort to build access features into a product, as early as possible (e.g., from its conception, to design and release). In the context of HCI, a proactive paradigm is advocated for the development of systems accommodating the broadest possible end-user population. In other words, design approaches are required that seek to minimize the need for a posteriori adaptations and deliver products that can be adapted for use by the widest possible end user population (adaptable user interfaces). This implies the provision of alternative interface manifestations depending on the abilities, requirements and preferences of the target user groups. The main objective in such a context is to ensure that each end-user is provided with the most appropriate

interactive experience at run-time. Producing and enumerating distinct interface designs through the conduct of multiple design processes would be an impractical solution, since the overall cost for managing in parallel such a large number of independent design processes, and for separately implementing each interface version, would be unacceptable [25].

As discussed above, the scope of design for diversity is broad and complex, since it involves issues pertaining to context-oriented design, diverse user requirements, as well as adaptable and adaptive interactive behaviors. This complexity arises from the numerous dimensions that are involved, and the multiplicity of aspects in each dimension. In this context, designers should be prepared to cope with large design spaces to accommodate design constraints posed by diversity in the target user population and the emerging contexts of use in the Information Society. Moreover, user adaptation it must be carefully planned, designed and accommodated into the life-cycle of an interactive system, from the early exploratory phases of design, through to evaluation, implementation and deployment. Additionally, design for diversity is anticipated to be an incremental process, in which designers need to invest effort in anticipating new as well as a changing requirements, and accommodating them explicitly in design through continuous updates.

## **2.6 User Interface Adaptation**

The Unified User Interface Development methodology ([25], [22], [23]) supports user interface adaptation through a process that leads to a single system that appropriately structures multiple designs and their underlying user and context related parameters, facilitating on the one hand the mapping of design to a target software system implementation; and on the other hand the maintenance, updating and extension of design itself.

The Unified User Interface Design method involves the identification of relevant design parameters, the design of alternative interface instances, the rationalization of a complex design space, and the final delivery of a complete interface adaptation logic. Polymorphic decomposition leads from abstract design pattern to a concrete artifact. Three categories of design artifacts may be subject to polymorphism on the basis of user- and usage-context- parameter values, and namely:

- **User tasks**, relating to what the user has to do; they are the centre of the polymorphic task decomposition process.

- **System tasks**, representing what the system has to do, or how it responds to particular user actions (e.g., feedback); in the polymorphic task decomposition process, they are treated in the same manner as user tasks.
- **Physical designs**, which concern the UI components on which user actions are to be performed; physical interface structure may also be subject to polymorphism.

User tasks, and in certain cases, system tasks, are not necessarily related to physical interaction, but may represent abstraction on either user- or system- actions. System tasks and user tasks may be freely combined within task “formulas”, defining how sequences of user-initiated actions and system-driven actions interrelate. The physical design, providing the interaction context, is associated with a particular user task, and provides the physical dialogue pattern associated to a task-structure definition. Hence, it plays the role of annotating the task hierarchy with physical design information.

Unified User Interface design emphasizes capturing of the more abstract structures and patterns inherent in the interface design, enabling hierarchical incremental specialization towards the lower physical-level of interaction, and making therefore possible to introduce design alternatives as close as possible to physical design. This makes it easier to update and extend the design space, since modifications due to the consideration of additional values of design parameters (e.g., considering new user- and usage-context- attribute values) can be applied locally to the lower-levels of the design, without affecting the rest of the design space.

Unified User Interface Design is recognized to require a higher initial effort and investment than traditional HCI design approaches. Tool-support including facilities for design re-use is considered particularly important in this respect. MENTOR [1] is a support tool for the process of Unified User Interface design, which provides Provision of practical integrated support for all the phases of conduct of the method by appropriately guiding the process and structuring the outcomes of creative design steps through appropriate editing facilities. However, MENTOR does not address the elaboration of appropriate interaction styles according to diverse target user and context characteristics. Thus design styles to be included in MENTOR design spaces need to be elaborated through other means, e.g., prototyping combined with guidelines compliance and user centered protocols.

### 3. Designing WUI Adaptation

As outlined in the previous section, existing design methods, techniques and tools in isolation are not sufficient to support adaptation design for the web. However they can be exploited in order to provide a suitable adaptation design framework.

In order to support WUIs adaptation, user-centered design, prototyping and guidelines can be used in combination with the Unified User Interfaces Design approach for designing web interfaces that can adapt to the diversity of the target user population in order to consolidate and make more easily available design adaptation knowledge.

The Unified Web Interfaces methodology [4] is derived from the architectural structure proposed for enabling the development of Unified User Interfaces [23], and is incarnated in the EAGER toolkit (see also the Chapter on “The development of web-based services” of this book), which supports its practical application and facilitates the design of WUIs adaptation. In particular, EAGER integrates a Design repository of:

- alternative primitive UI elements with enriched attributes (e.g., buttons, links, radios, etc.),
- alternative structural page elements (e.g., page templates, headers, footers, containers, etc.),
- fundamental abstract interaction dialogues in multiple alternative styles (e.g., navigation, file uploaders, paging styles, text entry).

The EAGER Designs Repository contains implemented and ready-to-use alternative elements (i.e., polymorphic task hierarchies) satisfying the requirements posed by specific user and context parameters values. These alternative styles have been designed following a User Centered Design approach combined with iterative prototyping and guidelines compliance. Additionally, EAGER design alternatives not only integrate current accessibility guidelines, but also provide a suitable approach to personalized accessibility [5]. In this respect, the EAGER Designs Repository can be viewed as encompassing consolidated adaptation design knowledge, thus greatly facilitating designers in the choice of suitable adaptations according to user-related or context-related parameters.

### 3.1 The EAGER Designs Repository

The Designs Repository component of EAGER provides the designs of alternative dialogues controls in a form of abstract design and polymorphism [22]. The following sections discuss concrete examples of the types of user interface adaptations that EAGER supports for web applications and services. For each alternative, a design and adaptation rationale is also reported which specifies the user and context-oriented parameters which are addressed.

#### 3.2.1 Adaptive Content

In the context of web applications and services, adaptation may involve the presentation of content in different forms according to various parameters.

An example is provided by images. Blind or low vision users are not interested in viewing images, but only in reading the alternative text that describes the image. In order to facilitate blind and low vision users, two design alternatives were produced which are presented in Figure 2. The text representation of the image simply does not present the image, but only a label with the prefix 'Image:' and followed by the alternative text of the image. The second representation, targeted to users with visual impairments, is same as the first with the difference that, instead of a label, a link is included that leads to the specific image giving the ability of saving the image. In particular, a blind user may not wish to view an image but may wish to save it to a disk and use it properly. In addition to the above, another design was produced that can be selected as a preference by web portal users in which the images are represented as thumbnail bounding the size that holds on the web page. A user who wishes to view the image in normal size may click on it. In Table 2, the design rationale of the alternative images design is presented.

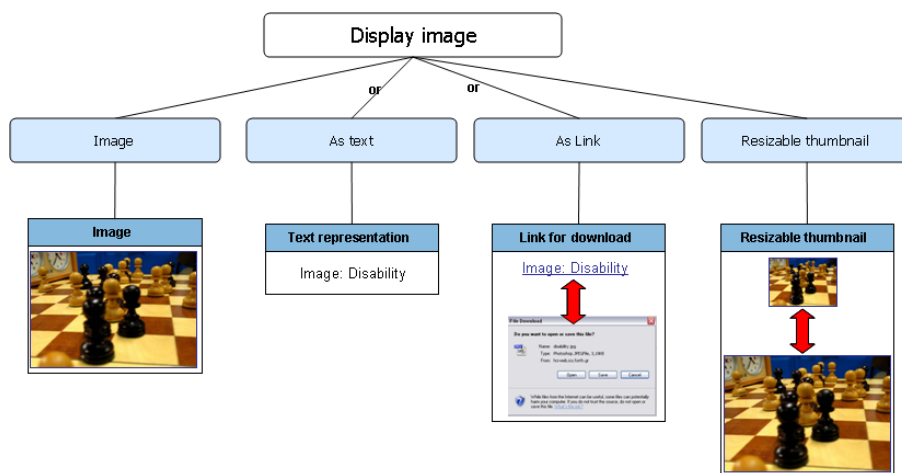


Figure 2. Image alternative representations

Table 2. Design rationale of the images alternatives

Task: Display image				
Style:	Image	As text	As link	Resizable thumbnail
<b>Targets:</b>	-	Facilitate screen reader and low vision users in order not to be in difficulties with image viewing	Facilitate screen reader and low vision users in order not to be in difficulties with image viewing but with the capability to save or view an image	Viewing images in small size in order not to hold large size on the web page with the capability to enlarge the image to normal size when it is necessitated.
<b>Parameters:</b>	User(Default)	User (Blind or Low vision)	User (Blind or Low vision) and user preference)	User (preference)
<b>Properties:</b>	View image	Read image alternative text	Read image alternative text or and select linked named as the image alternative text to save or view the image	View image thumbnail and select it to view it in normal size
<b>Relationships:</b>	Exclusive	Exclusive	Exclusive	Exclusive

Web based services usually also include lists of downloadable images. As presented in Figure 3, five alternative artifacts were designed according to user web expertise (Table 3). For novice users, images are presented as thumbnails, along with a link that downloads images and a description of the estimated time to download the image. For moderate users, the link is accompanied with the image size. Finally, an images list for expert users consists of the link to download the image along with the image name, size and type.

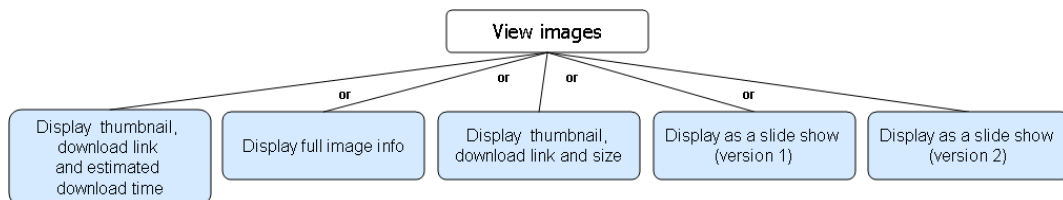


Figure 3. Images representations

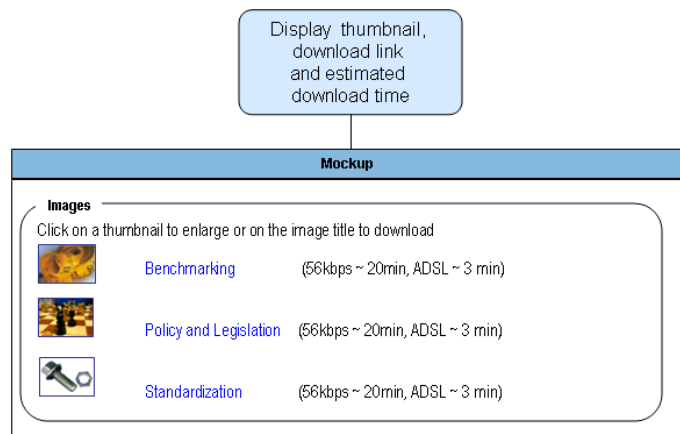


Figure 4. Display thumbnail, download link and estimated download time

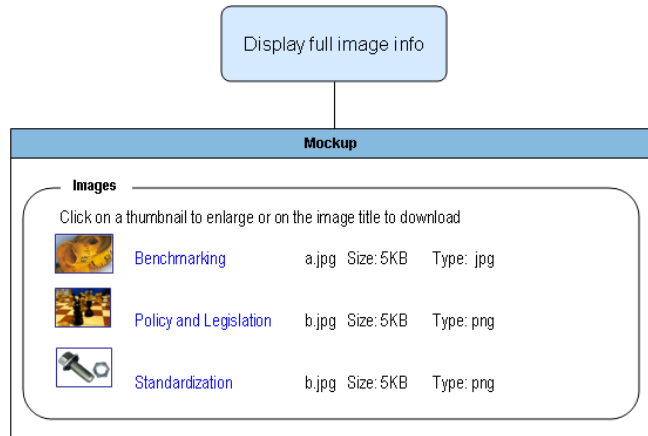


Figure 5. Display full image info

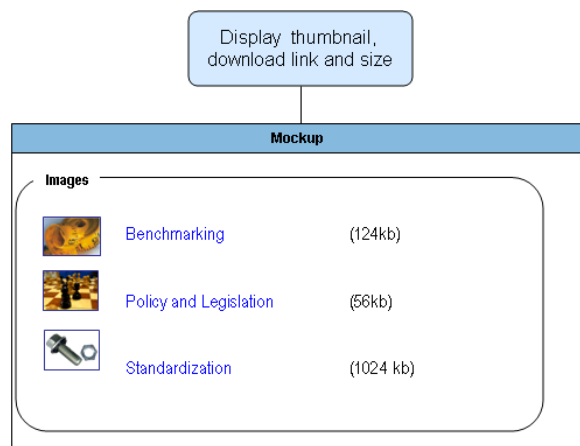


Figure 6. Display thumbnail, download link and size

In Figure 4, Figure 5 and Figure 6, images are presented as thumbnails, along with information varying according to user web expertise. In Figure 7 and Figure 8, a user who selects these representations may view images in a greater size and navigate among them by clicking on image buttons or on links respectively.

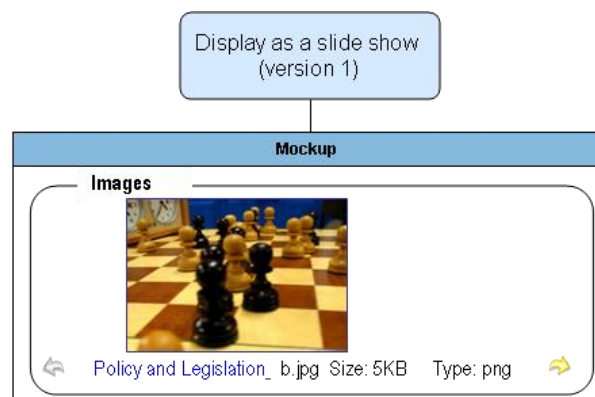


Figure 7. Display as a slide show (version 1)



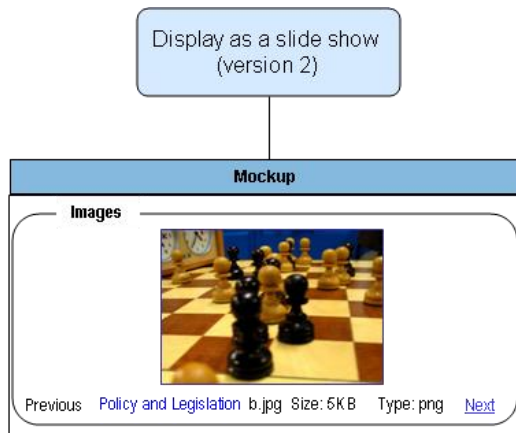


Figure 8. Display as a slide show (version 2)

Table 3. Design rational of images representations

Task: View images					
<b>Style:</b>	Display thumbnail, download link and estimated download time	Display full image info	Display thumbnail, download link and size	Display as a slide show (version 1)	Display as a slide show (version 2)
<b>Targets:</b>	Usability, flexibility	Usability, flexibility	Usability, flexibility	Usability, flexibility	Usability, flexibility
<b>Parameters:</b>	User (novice)	User (expert)	User (moderate)	User preferences	User preferences
<b>Properties:</b>	-	-	-	-	-
<b>Relationships:</b>	Exclusive	Exclusive	Exclusive	Exclusive	Exclusive

### 3.1.2 Adaptive Layout

A Web based service template generally maps to the generic scheme that incorporates the containers hosting contents. As presented in Figure 9, two generic template styles were designed. The linearized template style contains all the containers (top navigation, content, bottom navigation) in a linear form. On the other hand, the columns template style has three alternative styles where top and bottom navigation are placed on the top and bottom positions, and the centered container is split in two, three or four columns respectively for the two, three, four columns template.

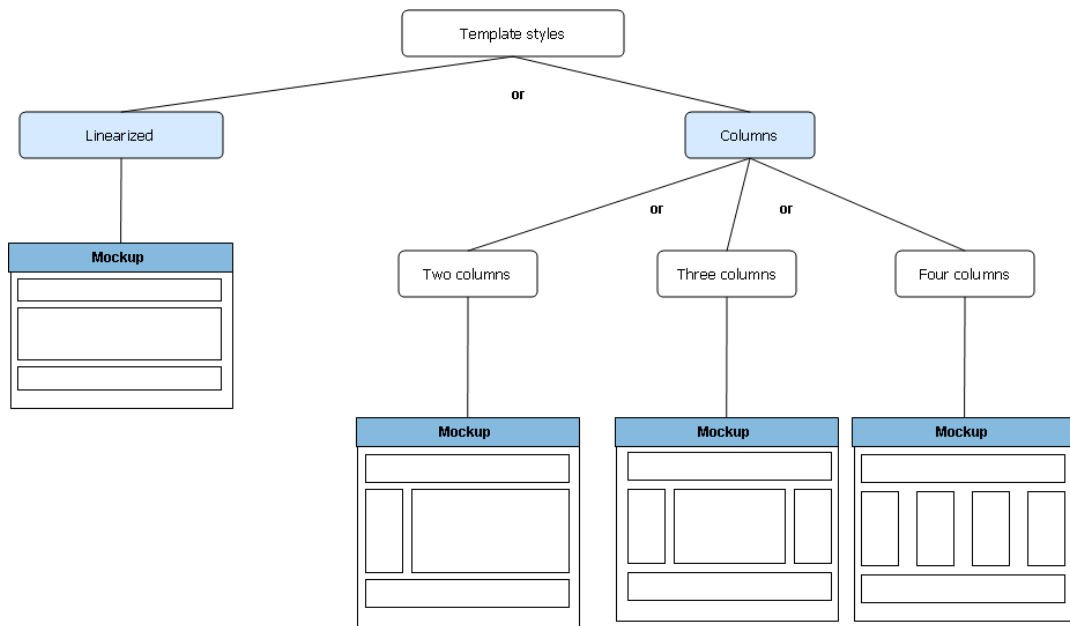


Figure 9. Template representation styles

According to the design rationale presented in Table 4, the linearized template supports speed, naturalness and flexibility for blind or low vision users, whereas the columns templates sustain speed, flexibility and optimum screen size for users with no visual impairments. The alternative columns templates are intended to be used in order to support content flexibility.

Table 4. Design rationale of the template styles

Task: Template styles		
<b>Style:</b>	Linearized	Columns
<b>Targets:</b>	Accessibility, speed, naturalness, flexibility	Speed, flexibility, cover optimum screen size
<b>Parameters:</b>	User (Blind, Low vision)	User (No visual impairments)
<b>Properties:</b>	-	-
<b>Relationships:</b>	Exclusive	Exclusive

Template size constitutes another significant aspect that is associated with the screen resolution in which the portal will be presented. According to [14], a web page has to be optimized for 1024x768 resolutions, but has to stretch well for any resolution, from 800x600 to 1280x1024 using a liquid layout. As presented in Figure 10 and Table 5, the template size may be resized according the device screen resolution in order to cover the optimum screen size.

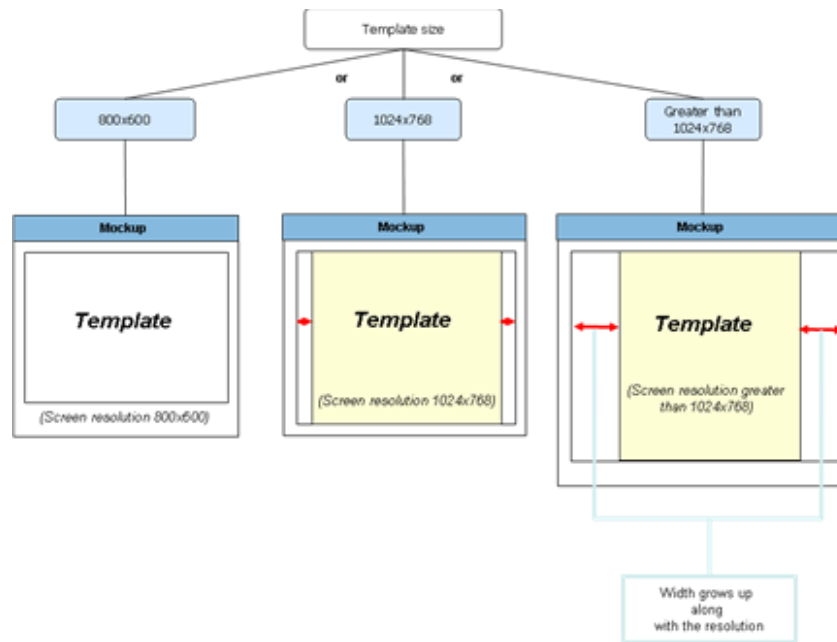


Figure 10. Template alternatives according to device resolution

When the resolution is 800x600, the template covers all the surface of the screen, whereas for 1024x768 resolutions the template has on its left and right sides a small unexploited area, in order to maximize the readability of the contents. For resolutions greater than 1024x768, the width of the empty areas on the left and right of the template is increased according to screen resolution.

Table 5. Design rationale of the templates alternatives according to device resolution

Task: Template styles			
<b>Style:</b>	800 x 600	1024 x 768	Greater than 1024 x 768
<b>Targets:</b>	Cover optimum screen size	Cover optimum screen size	Cover optimum screen size
<b>Parameters:</b>	Device resolution: 800 x 600	Device resolution: 1024 x 768	Device resolution: greater than 1024 x 768
<b>Properties:</b>	-	-	-
<b>Relationships:</b>	Exclusive	Exclusive	Exclusive

### 3.1.3 Adaptive Navigation

Navigation constitutes one of the main mechanisms that a web based service user uses. Multiple alternatives of the navigation mechanism were designed in order to support individual user abilities and preferences. These are presented in Figure 11, and the corresponding design rationale is presented in Table 6.

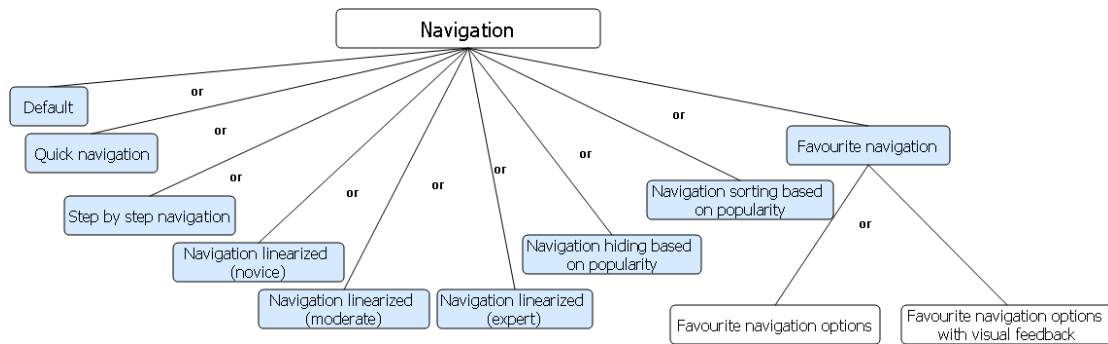


Figure 11. Navigation alternatives

The linearized navigation for novice users (see Figure 12) offers a linear form for all the navigation links of the portal, and in parallel a step by step navigation is supported. Initially, the user has to select among navigation hierarchies, next among entire navigation elements and finally among entire navigation sub-elements. In each step, the previous hierarchy is available in order to navigate back to another navigation hierarchy or navigation element. This step by step navigation mechanism offers a guided navigation to novice users with vision impairments, in order to enhance accessibility, flexibility and usability of the portal.

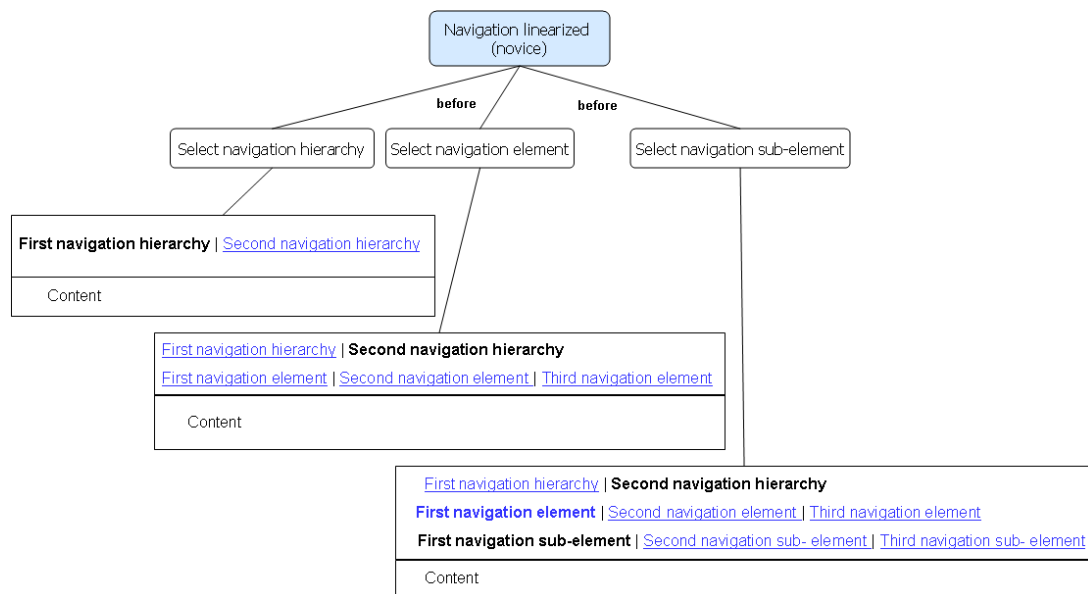


Figure 12. Navigation linearized (novice) alternative

The linearized navigation targeted for moderate with visual impairments supports a linearized form of the entire navigation of the portal. Initially, the user selects among navigation hierarchies and then the available navigation elements for the selected navigation hierarchy are presented, along with a navigation path through which the

user may navigate back to the navigation hierarchy. Through this procedure the user has to scan limited navigation options using the screen reader, knows each time which page are browsed, and always has an efficient way to navigate back to the navigation hierarchies thanks to the path mechanism (see Figure 13).

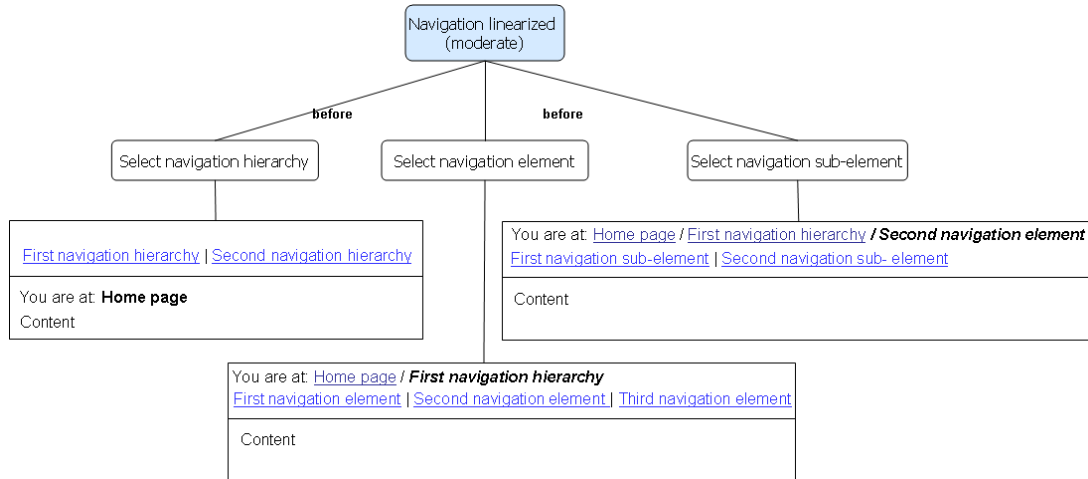


Figure 13. Navigation linearized (moderate) alternative

The linearized navigation for expert users with visual impairments resembles the linearized navigation for moderate users, but without the path mechanism. In this way, the expert has the ability to navigate back to the navigation hierarchy but is not notified about the web page browsed each time (see Figure 14). In this way, the expert can browse through the navigation mechanism quickly, without having the screen reader always reading the path of the entire page.

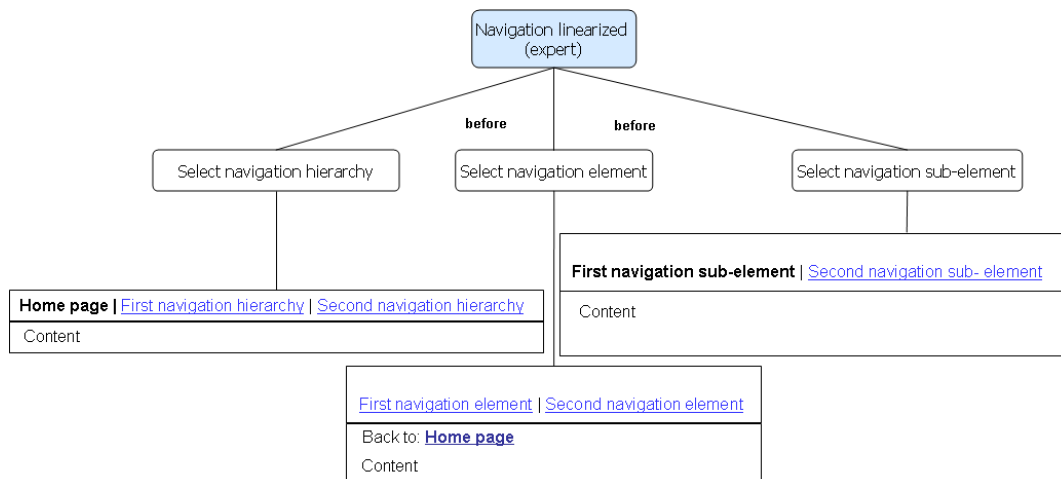


Figure 14. Navigation linearized (expert) alternative

Table 6. Design rationale of the step by step navigation – navigation linearized (novice) – navigation linearized (moderate) – navigation linearized (expert) alternatives

Task: Navigation (2/3)			
<b>Style:</b>	Navigation linearized (novice)	Navigation linearized (moderate)	Navigation linearized (expert)
<b>Targets:</b>	Accessibility, flexibility, usability	Accessibility, flexibility, usability, limited reading by the screen reader	Accessibility, speed, flexibility, usability, limited reading by the screen reader
<b>Parameters:</b>	User (Blind or Low vision and novice web expertise)	User (Blind or Low vision and moderate web expertise)	User (Blind or Low vision and expert web expertise)
<b>Properties:</b>	Navigation hierarchy first, navigation element next (for desired navigation hierarchy), navigation sub-element next (for desired navigation element)	Navigation hierarchy first, navigation element next (for desired navigation hierarchy), navigation sub-element next (for desired navigation element)	Navigation hierarchy first, navigation element next (for desired navigation hierarchy), navigation sub-element next (for desired navigation element)
<b>Relationships:</b>	Exclusive	Exclusive	Exclusive

### 3.1.4 Adaptive High level Interaction artifacts

Uploading files constitutes a frequently used function for web users. As shown in Figure 15 to Figure 18 and Table 7, three alternative designs were produced targeted to expert, moderate and novice users in order to upload and delete files.

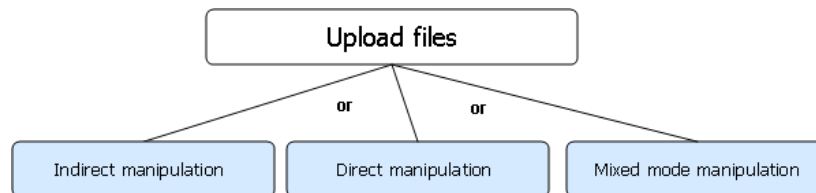


Figure 15. Upload files alternatives

As shown in Figure 16, a novice user in order to upload a file has to complete several simple steps; firstly, the user has to press button ‘add new’, then the interface changes, and the user has to complete three simple steps, browse a file, type a title and push the button ‘upload’. A progress bar with the file upload time appears. To delete an uploaded file, the user has to press button ‘delete’ and then to check the files to delete and press again the ‘delete’ button. The described presentation is targeted to novice users because it contains simple and detailed steps.

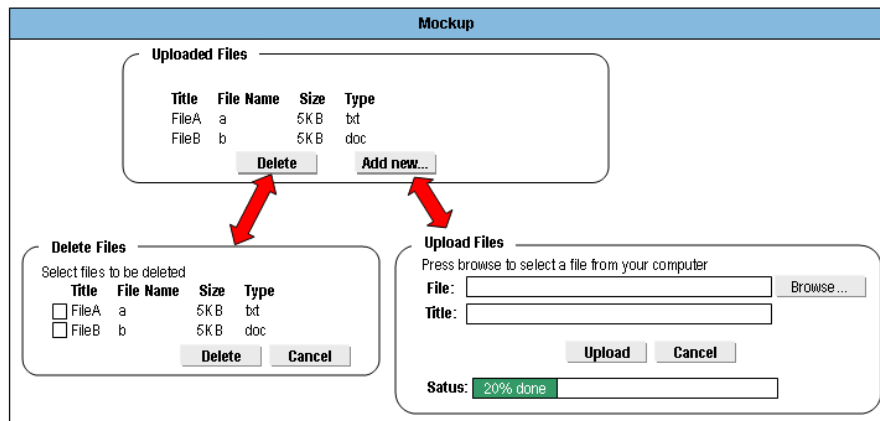
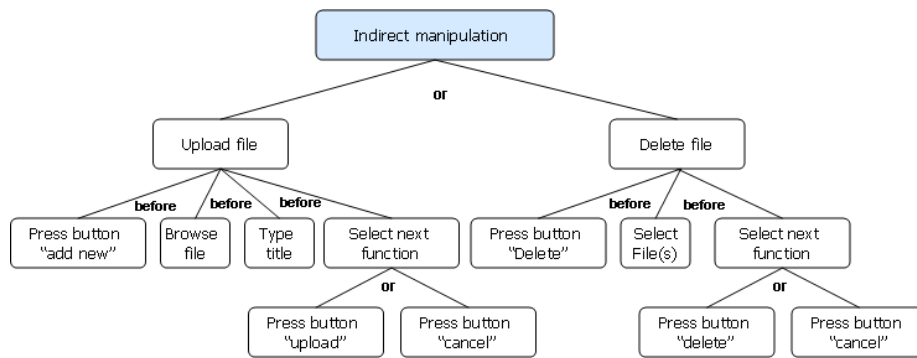
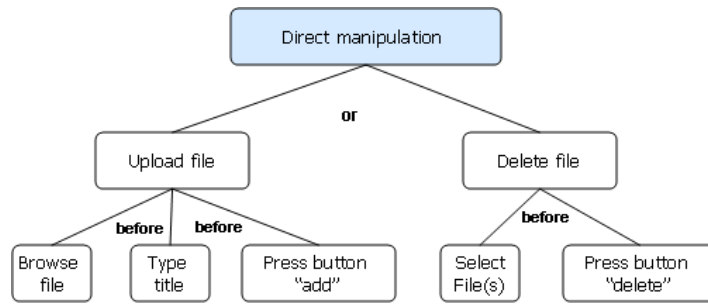


Figure 16. File upload indirect manipulation alternative

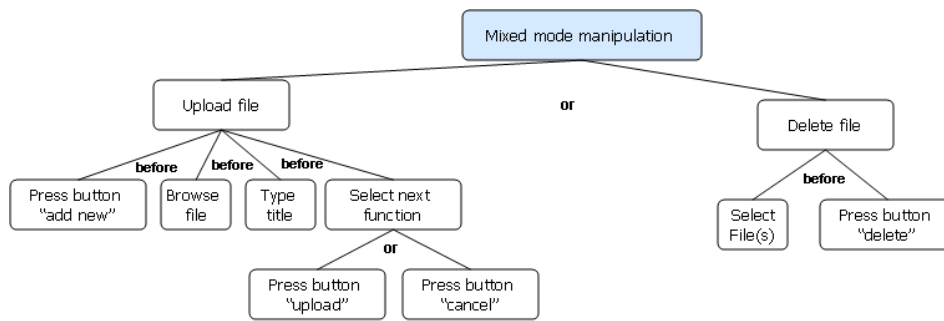
The direct manipulation representation (see Figure 17) is designed for expert users. All the functions are lying in a single interface in order to be accessed by the user quickly and effectively. The user has only to browse a file, type a title and press the button 'add' in order to upload a file. On the other hand, in order to delete a file that has already been uploaded, the user has only to select the file or files and press button 'delete'.

For moderate users, an intermediate design (between novice and expert user design) was prepared that includes two interfaces: one to upload files and another to view uploaded files and delete files these that were uploaded by accident (see Figure 18). The moderate user in order to upload a file has to press the button 'add', then automatically a second interface appears where the user browses a file, types a title and finally presses button 'upload'. A moderated user in order to delete a file uploaded by accident has only to check the file or files to be deleted and then to press the button 'Delete'.



	Title	File Name	Size	Type
<input type="checkbox"/>	FileA	a	5KB	txt
<input type="checkbox"/>	FileB	b	5KB	doc

Figure 17. File upload direct manipulation alternative



	Title	File Name	Size	Type
<input type="checkbox"/>	FileA	a	5KB	txt
<input type="checkbox"/>	FileB	b	5KB	doc

Figure 18. File upload mixed mode manipulation alternative



Table 7. Design rationale of the upload files alternatives

Task: Upload files			
<b>Style:</b>	Indirect manipulation	Direct manipulation	Mixed mode manipulation
<b>Targets:</b>	Simplicity, guided steps	Speed, effectiveness	Guides steps, effectiveness, usability
<b>Parameters:</b>	User (novice)	User (expert)	User (moderate)
<b>Properties:</b>	Upload file: Press button 'add new' first, browse file next, type title next, press button 'upload' Delete file: Press button 'delete' first, select file(s) next, press button 'delete'	Upload file: Browse file first, type title next, press button 'add' Delete file: Select file(s) first, press button 'delete' next	Upload file: Press button 'add new' first, browse file next, type title next, press button 'upload' Delete file: Select file(s) first, press button 'delete' next
<b>Relationships:</b>	Exclusive	Exclusive	Exclusive

Date entry constitutes another frequent task for web users in case of registering an event to the web portal or in case of searching for data based on dates. Five alternatives were designed for date entry, and are presented in Figure 19 and Figure 20. As recorded in Table 8, the first design includes three drop downs where user has to select three items (year, month, day) using dropdowns. These dropdowns are constructed in such a way that when a user selects a specific year and month, the appropriate calculations are made based on leap years and number of days that each month includes, in order that only the valid days are placed in the days' dropdown. This option is targeted to novice users because of the simplicity and error prevention that offers.

The second design looks like the first design, with the difference that the validation of the user input is made manually. The user after the selection of the appropriate values has to push the button 'ok' in order to validate the selected date. In case of a mistake, an error in red appears guiding the user to correct the mistake. This design artifact is targeted to users with visual impairments, since it does not necessitate extended screen reading and is simple in use.

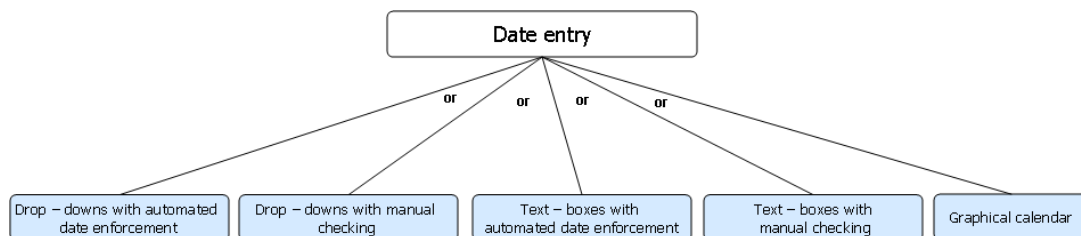


Figure 19. Date entry alternatives

As shown in Figure 20, in textboxes design with automated date enforcement design the user has to type year, month and day in the textboxes. The validation of the date

would take place when the user will try to complete the particular action. This date input alternative is designed for expert web users to improve the speed and the effectiveness of use. There is another design intended to be used by users who have visual impairments and are web experts too. Simply, in this design the user has to push the button 'ok' to validate the date inserted. For moderate and motor impaired users, another design was produced that offers a virtual calendar from where the user has monthly previews, may navigate through years and months using two dropdowns and can select a date by clicking on it when the appropriate date is met. This design is characterized as suitable for motor impaired users because it does not necessitate much user input, and is suitable for moderate users too because it offers a calendar-like graphical representation of dates that is more familiar to the users.

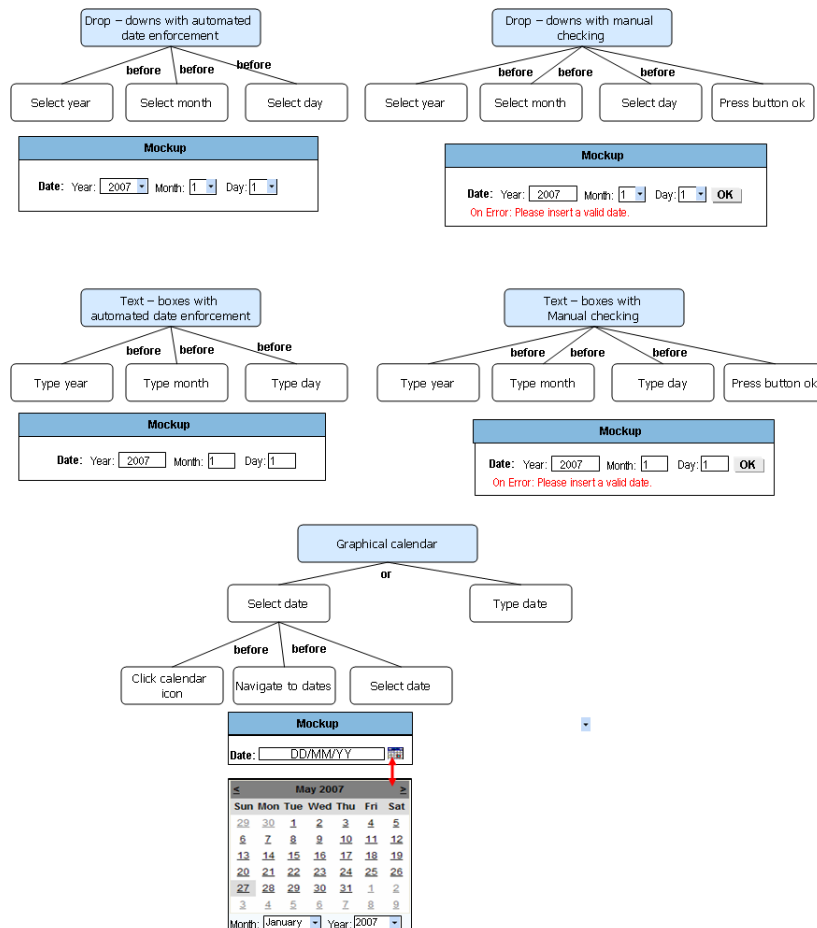


Figure 20. Date entry alternatives (In detail)

Table 8. Design rationale of date input alternatives

Task: Select date					
<b>Style:</b>	Drop downs with automated date enforcement	Drop down with manual checking	Text boxes with automated date enforcement	Text boxes with manual checking	Graphical calendar
<b>Targets:</b>	Simplicity, error prevention, easiness	Facilitate screen reader, effectiveness	Speed, effectiveness	Facilitate screen reader, effectiveness	Limited necessity of user input, simplicity
<b>Parameters:</b>	User (novice)	User (Blind or Low vision)	User (expert)	User ((Blind or Low vision) and expert)	User (Moderate, motor impaired)
<b>Properties:</b>	Select year first, month next and day at last	Select year first, month next, day next and press button 'ok' at last	Type year first, month next and day at last	Type year first, month next, day next and press button 'ok' at last	Type day first, '/' next, month next, '/' next, year at last or click on 'calendar' icon and select date on the virtual calendar
<b>Relationships:</b>	Exclusive	Exclusive	Exclusive	Exclusive	Exclusive

### 3.2 Designing UIs using the EAGER Design repository vs traditional UI prototyping

This section focuses on highlighting the benefits gained from using the approach followed by EAGER in the context of UI design in conjunction to the traditional prototyping techniques. This is accomplished both by providing an example of a simple interface (a web interface for posting a new discussion topic to a Message Board facility) as designed using the prototyping approach and its design in a task oriented fashion using the EAGER Designs Repository. Both processes are carried out by generating a page mockup. For the traditional prototyping approach Figure 21 presents a mockup that includes a title and a description field that are required along with the date entry module and a module for attaching files to topic. Files are attached using the browse button in order to locate the file, and the attach button to upload the located file. The field-set topic files are used to present the uploaded files. If a file was uploaded by mistake, the user can delete it by checking it and then by pressing the delete button.

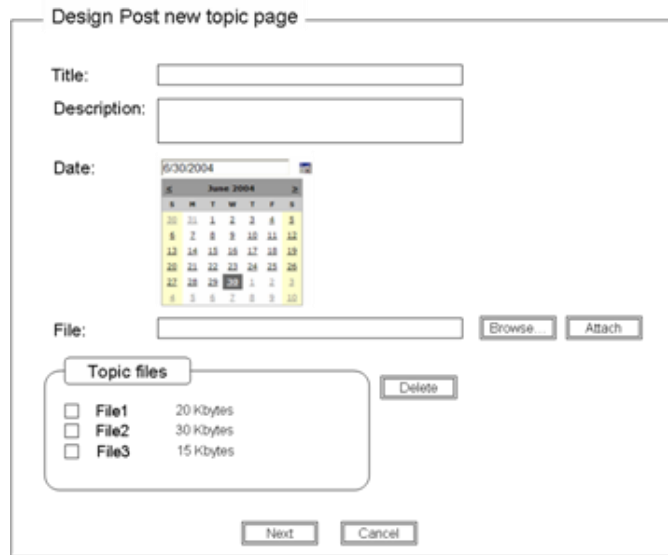


Figure 21. Mock-up of the page

The same design of a web page for posting topic demands different logic when it is intended to be developed with the EAGER toolkit. The design of such a web page is task oriented as long as the Designs Repository includes several alternative User Interface components which encapsulate functionality for several tasks. Figure 22 includes two text entry components, a date entry component, multiple files entry component and a component for functions to be applied.

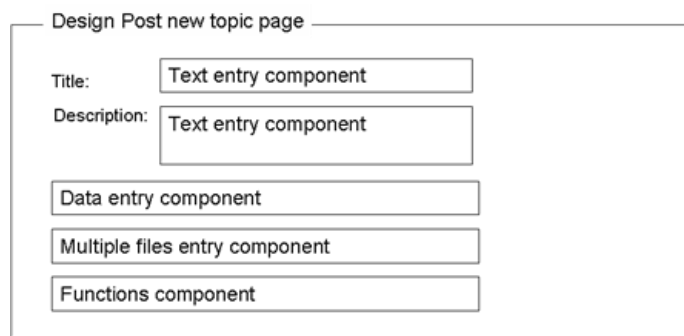


Figure 22. Task oriented mock-up of the page

A quick overview of these two page mockups can provide valuable information about their impact on the design process. In the EAGER based approach a designer has only to decompose each web-page in a number of user tasks supported by EAGER. In that way both the design demands less time from the part of the designer and also produces more solid designs as long as the supported user tasks are already designed and evaluated to support a number of alternative instantiation according to specific user and context requirements. Therefore the interfaces designed by means of EAGER are not monolithic but can have an exponential number of alternatives that are relative

to the combination of the user tasks selected for forming a web-page. An example of the resulting user interface as designed and developed by means of EAGER is presented in Figure 23. This figure presents two different instantiations of the designed user interface and highlights the following adaptations:

- Simple text entry controlled is adapted to a text entry component with an incorporated virtual keyboard.
- Graphical date selection dialog is adapted to a component that support date entry through selection of the appropriate values using three drop down menus.
- Graphical field-set is striped off its graphics and presented using its standard html form.
- Page functionality which is rendered as simple buttons is adapted to an alternative representation that offers additional help next to each function.

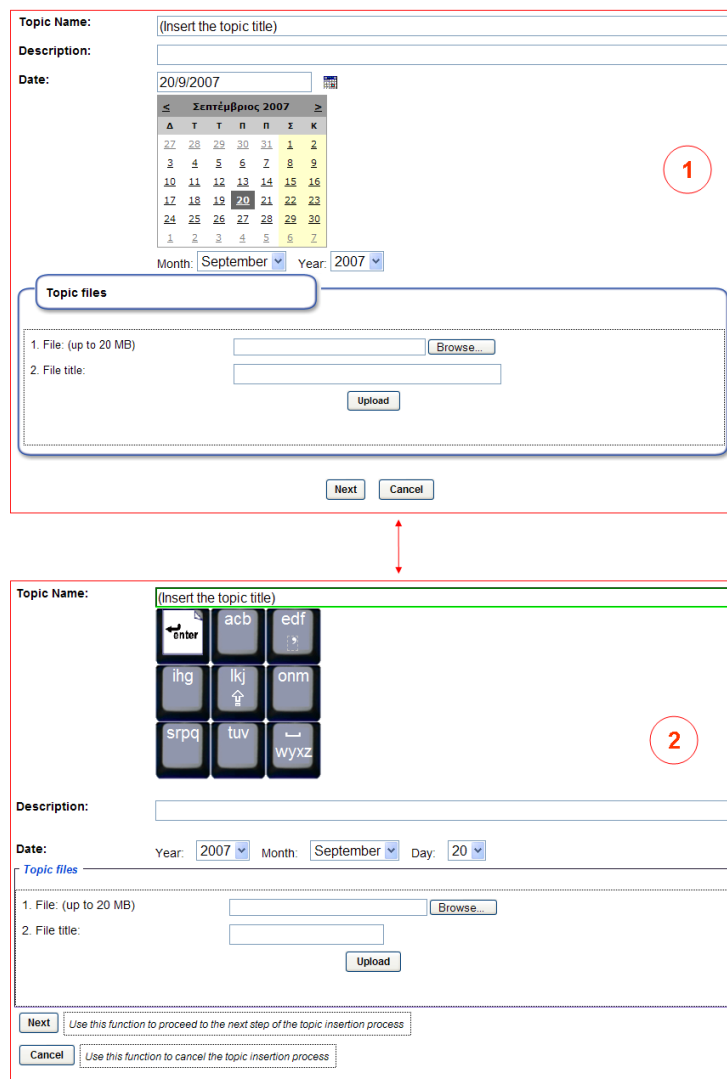


Figure 23. Post topic the alternative representations

## 4. Case Study: Design artifacts in a real life scenario

This section presents the way that the Designs Repository as implemented by EAGER can function in the context of a real life scenario. Toward this direction, the material presented here is functionality developed by means of EAGER in order to support the collaboration activities of the European Design for All e-Accessibility Network (EDeAN).

### 4.1 Adaptive Content

The way that images are displayed affects both the accessibility of a web user interface (e.g., for blind users) and its usability (e.g., in the case of large graphics with slow internet connection). It is, therefore, important to provide means for the conditional activation of images on the page template or content. These facilities were included in the designs repository and their use in a real life scenario is presented in Figure 24 and Figure 25 for the cases of page template and page content respectively.



Figure 24. Disable Graphics on template



Figure 25. Disable Graphics on page content

Additionally, the way that images are displayed can be determined by user selection in the context of EAGER. The available styles introduced in the designs repository allow the representation of images as (see Figure 26):

- Images
- their alternative text
- download link

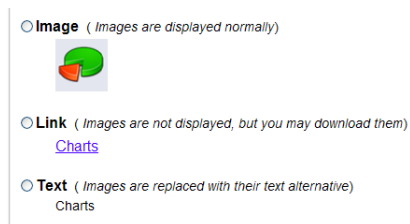


Figure 26. The image settings interface

## 4.2 Adaptive Layout

Skinning has become one of the major facilities offered for personalizing an interactive application according to the presentation preferences of a target user. The EDeAN case study incorporates an internal fully customizable skinning mechanism. This skinning mechanism can be manually invoked by the user or can be automatically used by the portal for adapting to several changes of context. An

example of manual skin selection for aesthetic purposes is presented in Figure 27 where the black skin is replaced by the crystal blue alternative.



Figure 27. Layout Change for aesthetic purposes

Skinning can also be used when the screen resolution used may cause inconsistencies in rendering the complete set of graphics contained in a web page. Such a case is presented in Figure 28, where a much simpler layout is automatically rendered when the screen resolution is changed from 1280x1024 to 800x600.



Figure 28. Layout adaptation based on device resolution (1280x1024 to 800x600)

The skinning mechanism can also be useful when changing the device used for accessing a web page, for example switching from a desktop pc with screen resolution



greater than 800x600 to a PDA with far lower screen resolution (320x240) (see Figure 29).



Figure 29. Layout adaptation based on device characteristics (PC to PDA)

### 4.3 Adaptive Navigation

This section aims at highlighting the ways that the alternative navigation schemes designed were integrated in the real life scenario of a web portal in order to facilitate different user characteristics and contexts of use. The default setting for a subscribed user is presented in Figure 30, introducing the commonly used mode of three different navigation menus located in the header, footer and left side of a web page.

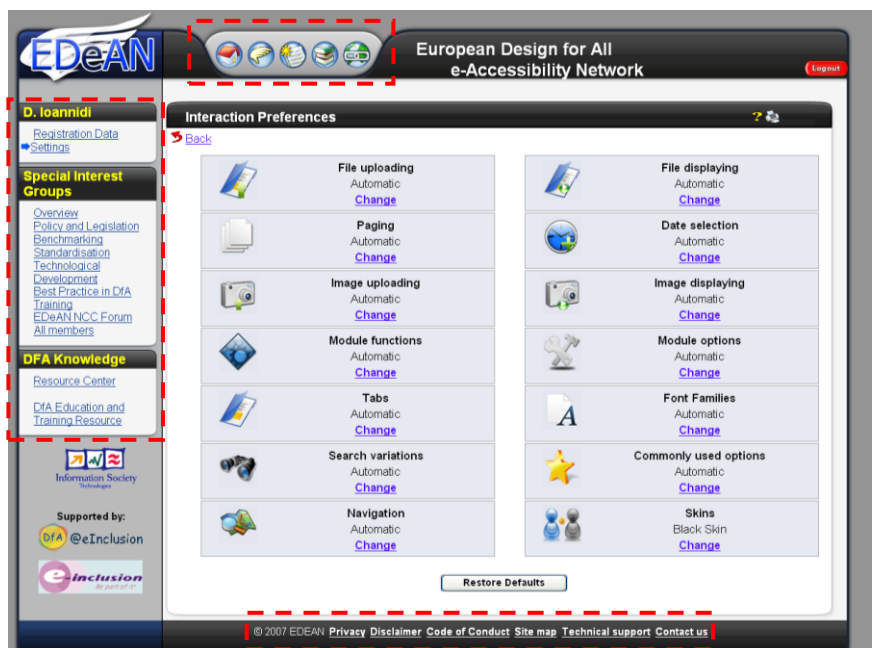


Figure 30. The default navigation scheme

An alternative navigation scheme is employed when the reduction of the total links displayed on a single web page is required (for example in the case of motor impaired users). This scheme, also called step by step navigation, is presented in Figure 31. In

this scheme only one submenu is activated when selected by the user, thus enabling the interface to render a minimal set of links.

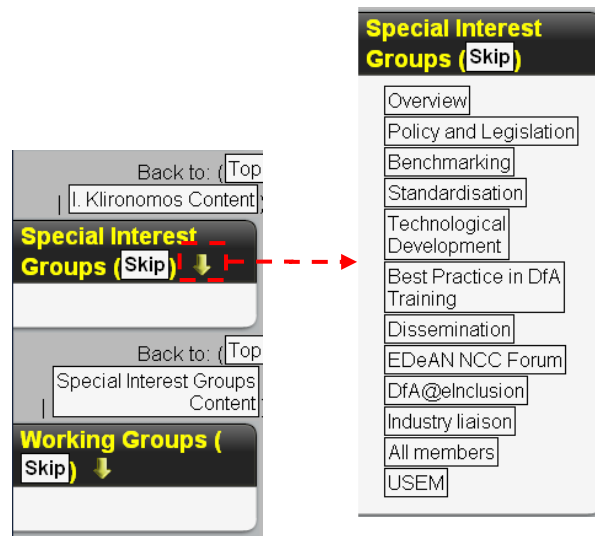


Figure 31. Step by Step navigation

The case of users with visual impairments introduces not only the need to display a minimum set of link selections but also minimize the need for spatial arrangement of menus. In that later case the default special arrangement of menus is not necessary because it may introduce difficulties in navigating to a web page. The scheme designed and implemented by EAGER unifies the available navigation menus and additionally introduces three alternative representations according to the familiarity of the user accessing the web site. This navigation scheme is presented in Figure 32.

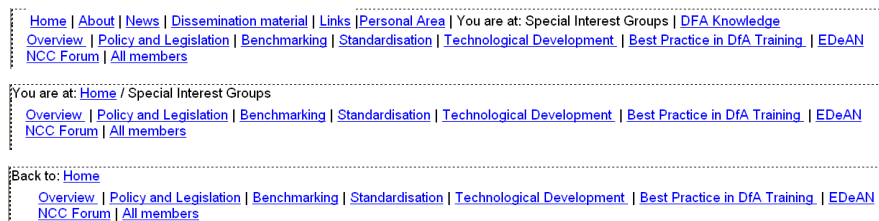


Figure 32. Single menu navigation

In the case where specific requirements are introduced by the device used for accessing the portal, such as for example when a PDA is used the navigation scheme is altered to support step by step navigation with an icon selection metaphor for offering a richer and therefore more flexible interaction. This navigation scheme used for presenting the portal on mobile devices is presented in Figure 33.

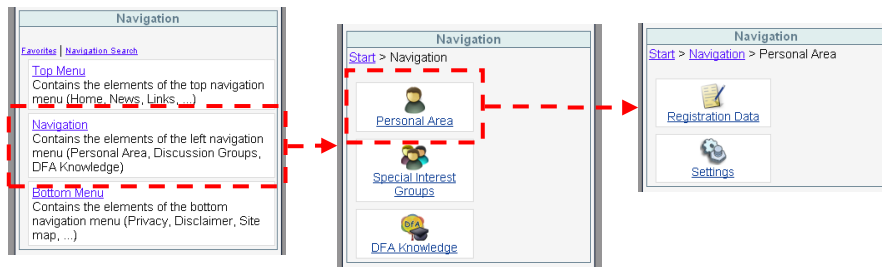


Figure 33. Step by step navigation with an icon selection metaphor

#### 4.4 Additional navigation facilities

In some cases where the need to easily identify the facilities most used by a user is important, some additional navigation facilities can be employed. In this context the most important feature designed and implemented in EAGER enables quick access to this facilities by either presenting them in a separate navigation window (see Figure 34) or highlighting them by a favorites icon (see Figure 35). Because the need for enabling these facilities cannot be directly linked with specific user or context characteristics, these facilities are enabled on user demand.

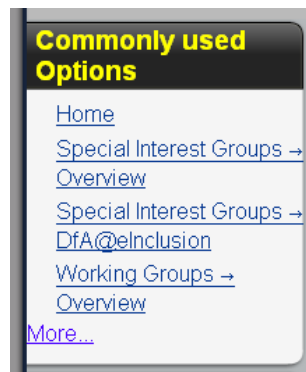


Figure 34. Favorite navigation options presented in a separate menu

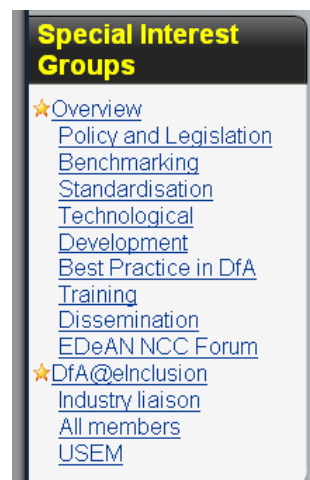


Figure 35. Favorite navigation options highlighted with a favorites icon

The same concept can be also supported by hiding the navigation options that are not commonly used. In the latter case, only these commonly user options are presented and user is provided with the option to expand this view to access the set of all available options as presented in Figure 36.

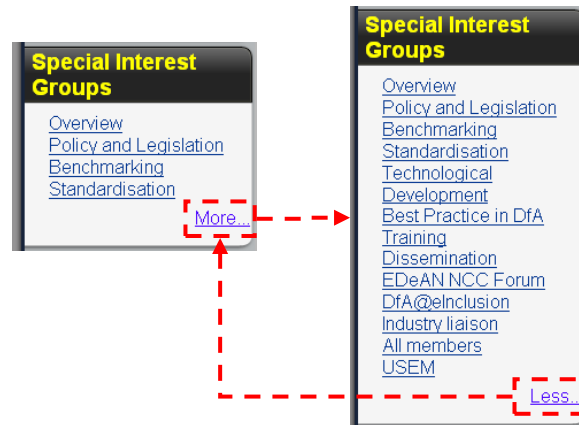


Figure 36. Navigation scheme where only the most commonly user options are presented

#### 4.5 Adaptive High level Interaction artifacts

The High level interaction artifacts introduced in the Designs Repository and developed in the context of EAGER represent high level tasks that can be carried out using different metaphors. This different instantiations of the same dialog can be in turn assigned to user characteristics and therefore conditionally activated on a specific user interface. This section highlights some example of high level interaction elements and their activation based on specific user characteristics. For example the direct invocation of the file uploading functions is activated for users with expertise on using the web while the indirect invocation help the separation of tasks which is helpful to users with low web familiarity. These two task variations are presented in Figure 37.

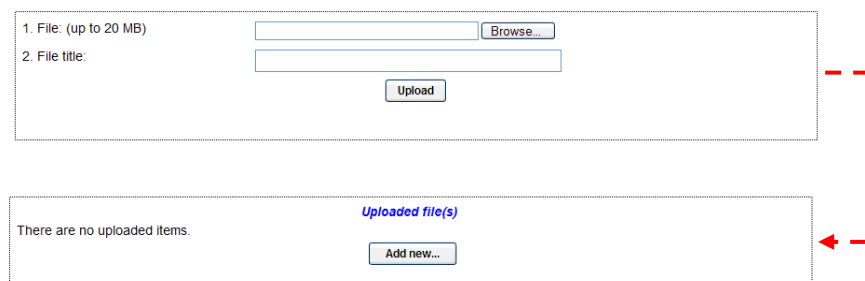


Figure 37. The File uploading style selection interface

Another high level element introduced by means of EAGER is the date selection dialog. Several different instantiations include direct selection of date using a

graphical calendar, manual selection using drop down menus or even manual date input by filling text fields (see Figure 38). These alternative styles can be assigned to users with low expertise (e.g., graphical) or for example blind users with high expertise on using the web (e.g., three dropdowns). Date entry affects the interactive elements presented to end users for date input. Towards this directions several interactions styles are incorporated such as the selection through a graphical calendar, the selection from drop down menus etc.

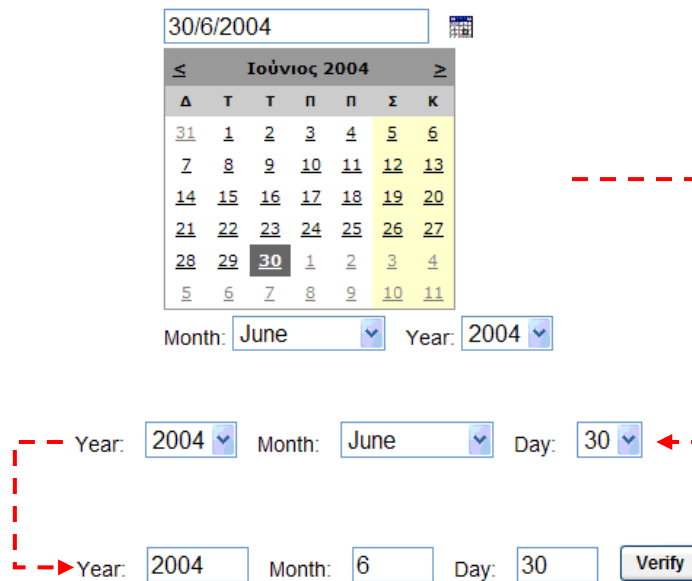


Figure 38. Alternative Date selection styles

Displaying a collection of images is a function that can have a great number of variations and no prominent selection. Towards this direction images can be displayed on a list with variations on the amount of information present or in a slide show version with navigation. Figure 39 presents the way a list of images can be rendered on a novice user (each image is presented together with the average download time) in conjunction with the way the same collection of images is presented to an expert user (each image is displayed together with full information such as size file name, etc.).

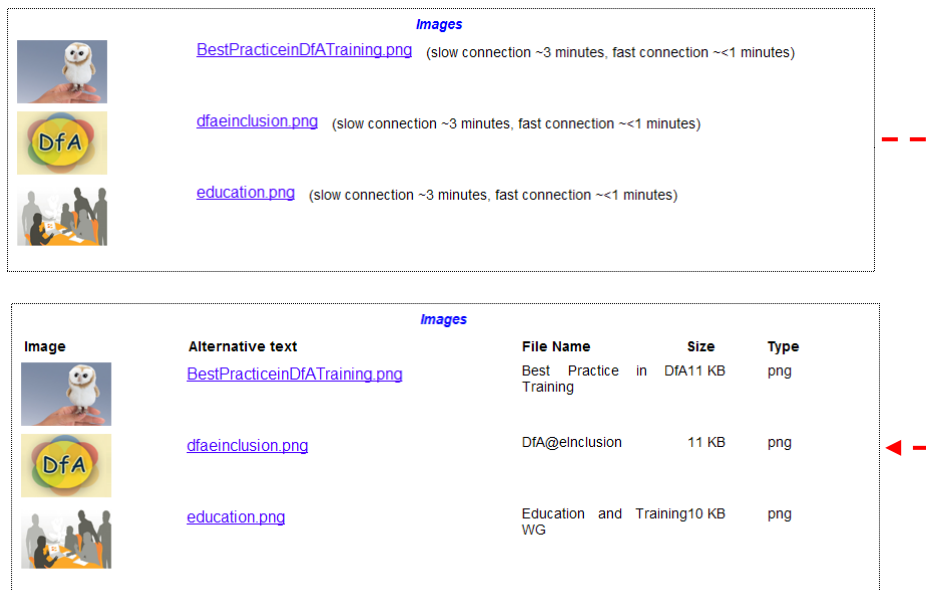


Figure 39. Alternative representations of a list of images based on user expertise

EAGER supports different representations for presenting list of images that do not necessarily correlate with specific user characteristics. As already mentioned the invocation of a specific style can be based on specific and direct user selection. In this context a presentation scheme for representing image lists is the so called slide show view presented in Figure 40.

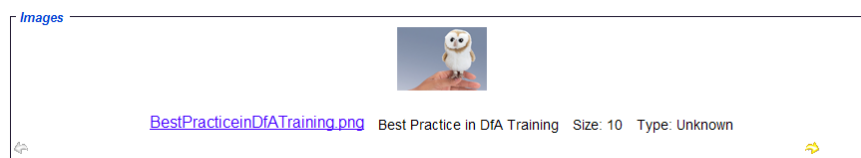


Figure 40. Presenting a collection of images as a slide show

## 5. Summary and Conclusions

This Chapter has presented a novel approach to the design of WUIs that is intended as an alternative to traditional design for the ‘average’ user and aims to ensure accessibility and usability for users with diverse characteristics. This approach is rooted in the tradition of user centered design, prototyping, and design guidelines, but also encompasses more recent trends towards user interface adaptation targeted to addressing the diversity of the end user population and producing WUIs able to adapt to specific user and context requirements. The EAGER toolkit incarnates such a design framework and facilitates web designers in adopting appropriate adaptations for various user and context-related parameters. To this purpose, EAGER makes easily available to designers a number of UI elements in various forms (polymorphic

task hierarchies) satisfying the requirements posed by specific user- / context-parameter values. Therefore, EAGER can reduce design effort and at the same time produce WUIs that are more accessible and usable since the selected abstract tasks are individually designed and evaluated to address the diverse need of their target users.

The EDeAN portal case study has made clear that the proposed approach allows embedding in Web-based applications automatic adaptation facilities for the benefit of accessibility and better user experience.

In the web context, however, the body of design knowledge regarding the appropriateness of design solutions to different (combinations of ) user- and context-related design parameters is to be considered as under continuous (re)elaboration due to the emergence of new requirements, contexts of use, and interaction technologies. The EAGER Designs Repository can be easily extended to include both additional alternatives for existing interactive artifacts and new interactive artifacts, thus allowing designers also to experiment and create their own adaptations.

Concerning additional enhancements of the EAGER toolkit, several advanced and intelligent techniques have been identified which can potentially improve its effectiveness and efficiency of the system. For instance, since the EAGER toolkit currently allows retrieving statistics on the designs preferred among various types of users, an advanced mechanism could be built-in in the future for allowing users to rapidly configure the layout and behavior of their personal interface according to what other users with similar profiles have selected.

On the other hand, all design elements can be assessed in terms of their conformance to the W3C accessibility guidelines and categorized according to their conformance level (i.e., single A, double A and triple A), thus allowing users (or developers) to select automatically which particular level of conformance they require, instead of selecting among predefined profiles and the assigned designs.

Another potential direction for future work concerns the integration of EAGER with a design support tool for the Unified User Interface Design method. For example, the MENTOR tool [1] provides practical support for all phases of Unified User Interface Design. By integrating in EAGER a web-based version of MENTOR, therefore, it will be possible to achieve a tool which graphically supports the development process of WUIs from design to implementation, and allows to easily and semi-automatically extend the EAGER user and context profiles and adaptation logic.

Finally, another potential extension is to render the EAGER toolkit a Web service so that Web developers using technologies other than .NET will be able to incorporate the EAGER adaptation logic into their artifacts, by providing an interface for defining profiles and receiving decisions regarding the activation – deactivation of alternative UI elements. Then, developers will only have to implement, if not available, the proposed alternative designs in their own development environments.

Overall, the approach presented in this Chapter is considered as a significant contribution towards embedding enhanced accessibility, usability and personalization in future and existing Web-based applications, and, ultimately, towards supporting individuals, especially people at risk of exclusion, to fully participate in the knowledge society.

## 6. References

- [1] Antona, M., Savidis, A., & Stephanidis, C. (2006). A Process–Oriented Interactive Design Environment for Automatic User Interface Adaptation. *International Journal of Human Computer Interaction*, 20 (2), 79-116.
- [2] Bonisch, B., Held, J., Krueger, H., (2003) Power Point for interface – simulation of complex machines. In Jacko J., Stephanidis C., *Human computer Interaction –Theory and Practice (Part II)*, Volume 2, Greece, (pp 1066 – 1070).
- [3] Dey Alexander Consulting. Email Usability.  
<http://deyalexander.com/resources/uxd/email-usability.html>
- [4] Doulgeraki, C., Partarakis, N., Mourouzis, A., & Stephanidis, C. (2009, to appear). Adaptable Web-based user interfaces: methodology and practice. *eMinds: International Journal on Human-Computer Interaction*.
- [5] Doulgeraki, C., Partarakis, N., Mourouzis, A. & Stephanidis, C. (2008). [A Development Toolkit for Unified Web-based User Interfaces](#). In K. Miesenberger, J. Klaus, W. Zagler, & A. Karshmer (Eds.), *Computers Helping People with Special Needs, Proceedings of the 11th International Conference, ICCHP 2008, 9-11 July 2008, Linz, Austria [LNCS 5105]* (pp. 346-353). Berlin Heidelberg, Germany: Springer.
- [6] Hackos, J., & Redisj, J. (1998) *User and Task Analysis for Interface Design*. New York: Willey (p. 376 )



- [7] Maguire M., (2001). Methods to support human-entered design, International Journal of Human-Computer Studies, Volume 55, Issue 4, (October 2001), pp. 587-634, Elsevier Science B.V.
- [8] Mariage, C., Vanderdonckt, J. & Pribeanu, C. (2005). State of the Art of Web Usability Guidelines, Chapter 41, in R.W. Proctor, K.-Ph.L. Vu (Eds), "The Handbook of Human Factors in Web Design", Lawrence Erlbaum Associates, Mahwah, pp-688-700.
- [9] Mayer, B.A., & Rosson, M.B. (1992) *Survey on user interface programming*. In Proc. of CHI'92, May 3-7, Monterey, USA.
- [10] Nielsen, J. (2000) *Designing Web Usability. The Art of the Simplicity*, New Riders Publishing, Indianapolis.
- [11] Nielsen, J., (2001). Search: Visible and Simple. Alertbox, <http://www.useit.com/alertbox/20010513.html>:
- [12] Nielsen, J. (2002) *Homepage Usability*, New Riders Publishing, Indianapolis.
- [13] Nielsen, J. (2003). Is Navigation Useful? Alertbox, <http://www.useit.com/alertbox/20000109.html>:
- [14] Nielsen, J. (2006) *Screen Resolution and Page Layout*. Nielsen, J., Alertbox. Available electronically at: [http://www.useit.com/alertbox/screen\\_resolution.html](http://www.useit.com/alertbox/screen_resolution.html).
- [15] Nielsen Norman Group Report Series. E-commerce User Experience. <http://www.nngroup.com/reports/ecommerce/>
- [16] Nielsen Norman Group Report. Site Map Usability. <http://www.nngroup.com/reports/sitemaps/>
- [17] Nielsen Norman Group Report. Web Usability for Senior Citizens. <http://www.nngroup.com/reports/seniors/>
- [18] Nielsen Norman Group Report. Usability of Websites for Children. <http://www.nngroup.com/reports/kids/>
- [19] Nielsen Norman Group Report. Beyond ALT Text: Making the Web Easy to Use for Users with Disabilities. <http://www.nngroup.com/reports/accessibility/>
- [20] Nielsen, J, Loranger, H. (). *Prioritizing Web Usability*. New Riders Press, Berkeley CA
- [21] Preece J., Maloney-Krichar D., (2003). Online communities: Focusing on sociability and usability. In J. Jacko & A. Sears (Eds.), *The Human-Computer Interaction Handbook – Fundamentals, Evolving Technologies and Emerging*

- Applications (pp. 596-620). Mahwah, New Jersey: Lawrence Erlbaum Associates (ISBN 0-8058-3838-4).
- [22] Savidis, A., Stephanidis, C. (2004). *Unified User Interface Design: Designing Universally Accessible Interactions*. International Journal of Interacting with Computers.
- [23] Savidis, A., Stephanidis, C. (2004). Unified User Interface Development: Software Engineering of Universally Accessible Interactions. *Universal Access in the Information Society*, 3 (3) (Managing Editor: Alfred Kobsa, University of California, Irvine, USA).
- [24] Savidis, A., Stephanidis, C., and Emiliani, P.L. (1997). *Abstract Task Definition and Incremental Polymorphic Physical Instantiation: The Unified Interface Design Method*. In G. Salvendy, M.J. Smith & R.J. Koubek (Eds.), *Design of Computing Systems: Cognitive Considerations [Proceedings of the 7th International Conference on Human-Computer Interaction (HCI International '97)]*, San Francisco, USA, 24-29 August (vol. 1, pp. 465-468). Amsterdam: Elsevier, Elsevier Science.
- [25] Stephanidis, C. (2001) *The concept of Unified User Interfaces*. In C. Stephanidis (Ed.), *User Interfaces for All - Concepts, Methods, and Tools*. Mahwah, NJ: Lawrence Erlbaum Associates, pp. 371-388 (ISBN 0-8058-2967-9).
- [26] Stephanidis, C. (Ed.), Salvendy, G., Akoumianakis, D., Bevan, N., Brewer, J., Emiliani, P.L., Galetsas, A., Haataja, S., Iakovidis, I., Jacko, J., Jenkins, P., Karshmer, A., Korn, P., Marcus, A., Murphy, H., Stary, C., Vanderheiden, G., Weber, G., & Ziegler, J. (1998) *Toward an Information Society for All: An International R&D Agenda*. *International Journal of Human-Computer Interaction*, 10 (2), 107-134.
- [27] Szekely, P., (1994) User interface Prototyping of Communicator Devices. *Interactions*. New Visions of Human – Computer Interaction. Volume IX. 6 Nov. – Dec. (pp 36-46).
- [28] The Rehabilitation Act Amendments (Section 508), <http://www.section508.gov/>
- [29] Vanderheiden, G., Chisholm, W., and Ewers, N. Design of HTML pages to increase their accessibility to users with disabilities, *Strategies for today and tomorrow*. Technical Report, Trace R&D Centre, University of Wisconsin - Madison, May 1996.

- [30] Vredenburg, K., Isensee, S., and Righi, C. (2002) User-Centered Design: An Integrated Approach. Upper Saddle River, NJ: Prentice Hall.
- [31] W3C. Web Accessibility Evaluation Tools: Overview.  
<http://www.w3.org/WAI/ER/tools/>
- [32] W3C. Web Content Accessibility Guidelines 1.0, W3C Recommendation, May 1999. <http://www.w3.org/TR/WCAG10/>.
- [33] W3C. Web Content Accessibility Guidelines 2.0, W3C Recommendation, December 2008. <http://www.w3.org/TR/WCAG20/>
- [34] W3C. Mobile Web Best Practices 1.0. Mobile Web Best Practices 1.0. Working Draft 17 October 2005
- [35] Zhu, W., Vu, K.-P. L., & Proctor, R. W. (2005). Evaluating Web usability. In R. W. Proctor & K.-P. L. Vu (Eds.), Handbook of human factors in Web design (pp. 321-337). Mahwah, NJ: Erlbaum.