The accessibility of web applications: the case of portals and portlets

Evangelos Vlachogiannis, Jenny Darzentas, Argyris Arnellos, Thomas Spyrou, John Darzentas

University of the Aegean, Department of Product & Systems Design Engineering
84100, Ermoupolis - Syros, Greece
{evlach, jennyd, arar, tsp, idarz }@aegean.gr

Abstract

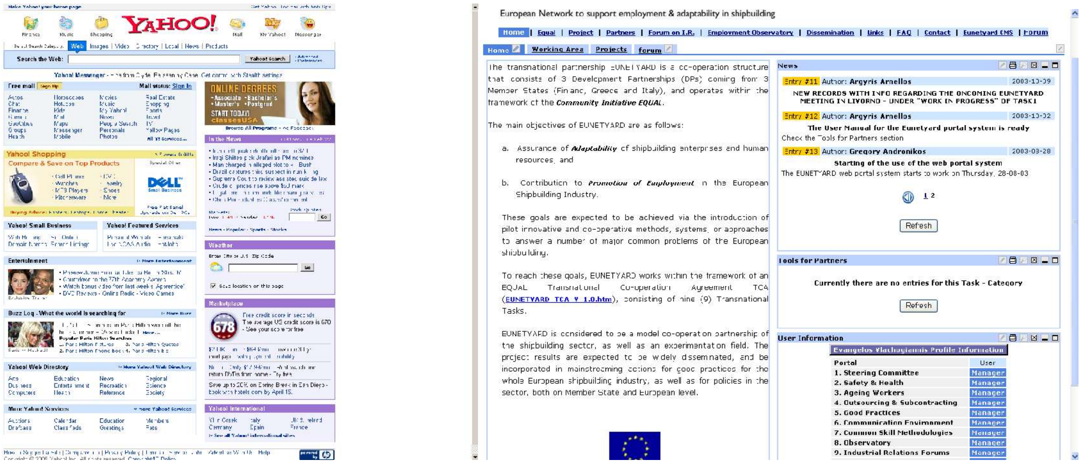
This paper discusses the abstraction and extension of portal systems and definitions of portlets in order for them to become more accessible. A whole/parts perspective is adopted that considers portal systems as a whole and the portlets as parts that need to have some attributes, behaviours and organisation for accomplishing their purpose. A comparison between a portal system and a web site in terms of accessibility is given for identifying the additional requirements of a portal system. According to these, complementary specifications and guidelines are proposed at the design as well as at the implementation level, and suggestions are made for some types of automatic accessibility checking for portals.

1 Introduction

Nowadays the demand is for most applications to be web based. The web has proven to be much more than a service, instead it has become an accepted way of offering services in many areas, such as communication, education, entertainment, etc. It has surpassed its original design goals and, as the ways to access the Internet proliferates (along with the range of services offered), it becomes increasingly important to ensure accessibility of these services. Nowhere is this more apparent than in the case of people with disabilities. Many of these people have problems accessing the Internet, yet paradoxically, these are the people who stand to benefit greatly from the opportunities offered by it for online services that encourage independent living.

From a content management point of view, as the functionality and management of web applications become more and more complex, companies have been motivated to invest in portal systems as a mechanism by which they can manage information in a cohesive and structured fashion [Bellas et al. 2004]. Portals are used in enterprises both as their communication link with their customers, or maybe resellers, and often also as their intranet online management information tool. This also applies to education [U-Portal],[PORTAL], where there is a tendency from institutes to develop integrated and shared portal frameworks for moving all their services, back office operations and even teaching content online.

Portals may be divided into "horizontal" and "vertical": the former refer to those that contain a huge variety of thematic content like Excite and Yahoo; whereas vertical portals are those that serve a more specific and narrower scope and provide knowledge in greater depth . For instance, in [EQUAL – SYMPOLITIA] a portal infrastructure is being used as a virtual incubator for communication and learning between educators, consultants and interested parties, while in [EQUAL-EUNETYARD] it is used as the information and communication tool between the project's partners. An example of both categories can be viewed in Figure 1.



The transnational partnership EUNETYARD is a co-operation structure that consists of 3 Development Partnerships (DPs) coming from 3 Member States (Financ, Greece and Italy), and operates within the framework of the ***Community Initiative EQUAL***.

The main objectives of EUNETYARD are as follows:

a. Assurance of ***Adaptability*** of shipbuilding enterprises and human resources, and

b. Contribution to ***Promotion of Employment*** in the European Shipbuilding Industry.

These goals are expected to be achieved via the introduction of pilot innovative and co-operative methods, systems, or approaches to answer a number of major common problems of the European shipbuilding.

To reach these goals, EUNETYARD works within the framework of an EQUAL Transnational Co-operation Agreement TCA (EUNETYARD_TCA_V_1.0.htm), consisting of nine (9) Transnational Tasks.

EUNETYARD is considered to be a model co-operation partnership of the shipbuilding sector, as well as an experimentation field. The project results are expected to be widely disseminated, and be incorporated in mainstreaming actions for good practices for the whole European shipbuilding industry, as well as for policies in the sector, both on Member State and European level.

Entry #11 Autor: Argyris Arnellos 2003-10-09

NEW RECORDS WITH INFO REGARDING THE ONCOMING EUNETYARD MEETING IN LIVORNO - UNDER "WORK IN PROGRESS" OF TASK1

Entry #12 Autor: Argyris Arnellos 2003-10-02

The User Manual for the Eunetyard portal system is ready

Check the Tools for Partners section

Entry #13 Autor: Gregory Andronikos 2003-03-28

Starting of the use of the web portal system

The EUNETYARD web portal system starts to work on Thursday, 28-08-03

1 2

Currently there are no entries for this Task - Category

Evangelos Vlachogiannis Profile Information

Portal	User
1. Steering Committee	Manager
2. Safety & Health	Manager
3. Ageing Workers	Manager
4. Outsourcing & Subcontracting	Manager
5. Good Practices	Manager
6. Communication Environment	Manager
7. Common Skill Methodologies	Manager
8. Observatory	Manager
9. Industrial Relations Forums	Manager

Figure 1 : (left) A horizontal portal (http://www.yahoo.com), (right) A vertical portal (http://www.eunetyard.net)

It is possible to draw a comparison between portals and Graphical User Interfaces (GUIs), in that portals provide a uniform way of accessing applications and content on the web, just as GUIs provided to operating systems. Or, alternatively portals could be seen as operating systems that provide a basic functionality and on which portlet applications can be based (according to certain specifications). In this way, just as Java provided a virtual machine to satisfy interoperability, the portal's developers created the portlet specifications and their implementations (portlet containers) that handle a portlet's life cycle.

There are many definitions for "portal" [Smith, 2004] and very many web pages contain in the heading the term "portal" but the formal definition of portal given by the Java Community Process in the Java Specification Requests (JSR 168) Portlet Specification [JCP] states that:

> "*A portal is a web based application that –commonly- provides personalisation, single sign on, content aggregation from different sources and hosts the presentation layer of Information Systems. As part of a portal, there may be sophisticated personalisation features for providing customised content to users. Portal pages may have different sets of portlets creating content for different users. A portlet is a Java technology based web component managed by a portlet container, which processes requests and generates dynamic content. Portlets are used by portals as pluggable user interface components that provide a presentation layer to Information Systems.*"

This, of course, is a definition from a software engineering perspective. This paper tries to provide a more generic definition in order to approach the problem from a human – computer interaction (HCI) and universal design (UD) approach.

The accessibility of portals has been investigated [Gappa & Nordbrock, 2004], and the research concluded that portals are not sufficiently accessible and this is due both to their functional and managerial complexity. By "web accessibility" it is considered that ideally, both hypertext or multimedia structure and content will be communicated in an accessible way. Accessibility is defined as the ability of an actor to access a resource fruitfully where an actor is defined as an entity containing an instance of a user, the user's access media and the user's context of use. An accessible system is a system designed with the aim to be as accessible, context suitable and usable to anyone through their preferred access media, as far as is possible. This means that an accessible system has to be aware of actor capabilities and preferences in the current application context.

A web page hypertext structure, compared with that of portals, can be much more easily accessible following the Web accessibility guidelines and Techniques[1]. On the other hand, simple web pages often do not offer adaptation according to the actor. A critical parameter of accessibility that becomes even more acute due to the portal's complexity, is navigation. Additionally, the number of people needed to be involved for maintaining a portal

[1] Web Content Accessibility Guidelines 2.0 - W3C Working Draft 19 November 2004 (http://www.w3.org/TR/WCAG20/)

increases, while often their computing background differs greatly. Keeping in mind that a portal can also contain fragments of more than one web technology, controlling accessibility of a portal is a difficult and multifaceted task.

This paper proposes the abstraction and extension of portal systems and definitions of portlets in order for them to become more accessible. A whole/parts perspective is taken that considers portal systems as a whole and the portlets as parts that need to have some attributes, behaviours and organisation for accomplishing their purpose. If one adds the capabilities of the device that the user makes use of to access the web (e.g. text browser, mobile phone, etc) also known as user agent by the W3C [2] then the problem space might be expressed as:

Accessible web = accessible parsing of (accessible web objects + accessible organisation of the web objects)

Given this expression then it follows that an accessible portal implies the need for accessible portlets. Furthermore, the dimension of accessibility should be extended to the portlets' positioning in the portal. Accessible positioning of the portlets for the user could include the implementation of accessible navigation, or even a new way of communicating the portlets to the user (i.e. mapping portlets to a kind of accessible page with navigation). At a more applied level, accessible portlets could be developed by extending the portlet specifications towards more accessible schemata. The accessibility factor in the case of the positioning of the portlets can primarily be improved by the adaptation of current accessibility guidelines within the framework of the portal and of the respective positioning mechanisms. Finally, other characteristics of portals could also be considered as a way to enhance their accessibility, for instance, content syndication could be used as a means of summary extraction of the required content specifically for cognitively disabled users.

2 Special requirements for portals and some proposals for dealing with them

The abovementioned portal and portlet definitions need to be generalised and abstracted so that accessibility principles can be applied. Portal systems are defined as a web based desktop environment providing a dynamic set of application and content interfaces servicing the goals of the portal provider. The description "web based desktop environment" emphasises both the integrated - personalised properties and the collaboration features of portals that can be provided to online communities. Such applications can be links to other sites, e-mail, forums, search engines, on-line shopping malls, chat rooms, filtering and blocking options for parents, a wide range of content, services and vendor links, news, reference tools, white and yellow pages directories, discussion groups, online shopping and content management subsystems and many more.

The portal provider can offer both content and application interfaces according to purpose and user. These interfaces are the portlets' interfaces. Portlets constitute the whole architecture of an application including the interfaces appearing on a portal. These interfaces need to follow some specifications for being aggregated to a whole page. Portlet applications consist of a group of target related portlets interfaces.

In the rest of this section an attempt to classify the special requirements of portal systems that need to be met from an accessibility point of view is undertaken, principally by comparing with them with simple web pages.

2.1 Content portlets / subsystems

Portlet interfaces may consist of hypertext and/or multimedia content. To be accessible, these need to follow web accessibility guidelines, just as for web pages. The W3C's WAI Web Content Accessibility Guidelines (WCAG)' is currently at a "working draft" status of second version and they are based on the following basic principles:

- Content must be perceivable,
- Interface elements in the content must be operable,
- Content and controls must be understandable,

[2] W3C Glossary (http://www.w3.org/TR/WAI-USERAGENT/glossary.html#u): In this document, the term "user agent" is used in two ways:

- The software and documentation components that together, conform to the requirements of this document. This is the most common use of the term in this document and is the usage in the checkpoints.
- Any software that retrieves and renders Web content for users. This may include Web browsers, media players, plug-ins, and other programs — including assistive technologies — that help in retrieving and rendering Web content.

- Content must be robust enough to work with current and future technologies.

Here a distinction between portlets and web pages needs to be made because of the former's "page fragment" nature. The requirements for accessible portlets can be described on two levels: as principles, or guidelines, and at the level of technology. The principles are general and technology independent, whereas the technology level needs to be adapted to each technology, e.g.html, wml, etc.

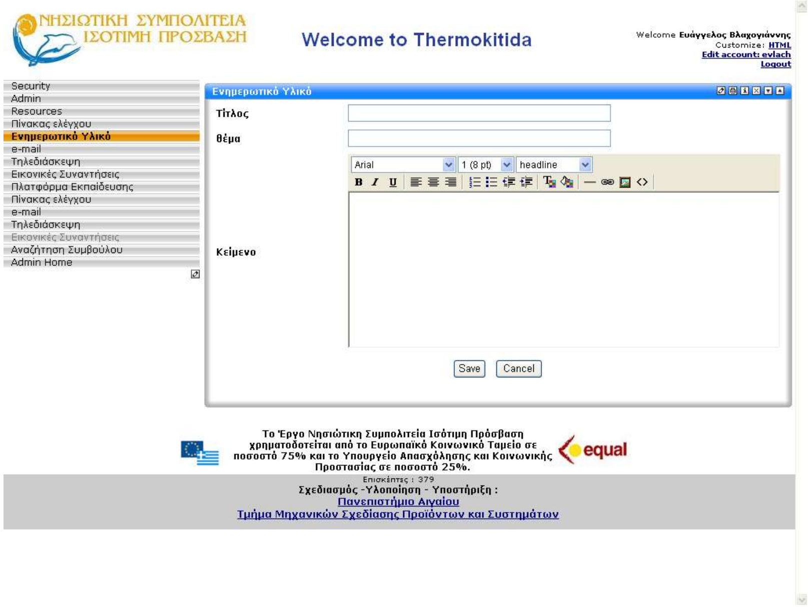
It should be mentioned that this attempt does not claim to be an exhaustive investigation of the requirements, but aims to show up the need for such work by an organisation like the W3C WAI and thus tries to provide a roadmap. What follows is indicative work extracted from the authors' experience on portals / accessibility [BenToWeb], [IRIS], [EQUAL - EUNETYARD], [EQUAL - NEORIO2] and best practices [Hepper & Lamb, 2004] that might apply to static or dynamically generated html markup.

- *Principle 1***:** If there is not special portlet requirement, follow Web Content Accessibility Guidelines
- *Principle 2***:** Global page information is not permitted.
  - According to [HTML] specification for a page to be valid it needs to contain only one of the header tags. So, do not use the following tags : <html>, <head>,<title>,<meta>, <link>, <body>
  - Avoid client scripts or use with caution.
- *Principle 3***:** The presentation of each portlet needs to be separated from the content and use portlet style classes instead of style-oriented attributes.
  - Do not use formatting attributes. Use [CSS] instead.
  - Use portlet style classes as described in JSR-168.
- *Principle 4***:** Remain in portal environment and on working portlet.
  - Do not use pop-up windows unless it is about an external link.
  - Do not use <iframe> unless you provide alternative content.
  - Do mark a beginning of a portlet instance by using a anchor so that the portal system can directly point user to her working portlet instance. (e.g. use id="portletbookmark-<portlet instance>")
- *Principle 5***:** Provide alternative modalities and languages
  - Additional modes (like view and customise) can be added to the portlet specifications to include alternative modalities of both content or/and interaction of portlets.
  - Use internationalisation mechanism of your portal system.

Following such principles and techniques becomes more difficult as the number of portlets increases and they originate from providers with varying background knowledge. It is neither very feasible to separate the information providers from the editor nor to educate them on accessible portal design. The best solution is possibly a "What You See What You Get" editor, as shown in Figure 2, with controlled functionality, so that, for instance, an author cannot change the font style but can underline and automatically add needed portlet related markup as described above. This editor would need to follow the Authoring Tool Accessibility Guidelines[3] (e.g. an uploaded image will require an alternative text). These guidelines include:

- Support accessible authoring practices,
- Generate standard markup,
- Support the creation of accessible content,
- Provide ways of checking and correcting inaccessible content,
- Integrate accessibility solutions into the overall "look and feel",
- Promote accessibility in help and documentation
- Ensure that the authoring tool is accessible to authors with disabilities.

[3] Authoring Tool Accessibility Guidelines 1.0 - W3C Recommendation 3 February 2000 http://www.w3.org/TR/WAI-AUTOOLS/



Welcome Ευάγγελος Βλαχογιάννης
Customize: HTML
Edit account: evlach
Logout

Arial
1 (8 pt)
headline

Το Έργο Νησιώτικη Συμπολιτεία Ισότιμη Πρόσβαση χρηματοδοτείται από το Ευρωπαϊκό Κοινωνικό Ταμείο σε ποσοστό 75% και το Υπουργείο Απασχόλησης και Κοινωνικής Προστασίας σε ποσοστό 25%.

Επισκέπτες : 379
Σχεδιασμός -Υλοποίηση - Υποστήριξη :
Πανεπιστήμιο Αιγαίου
Τμήμα Μηχανικών Σχεδίασης Προϊόντων και Συστημάτων

Figure 2: Provide content using online forms with WYSWYG editors

Of course this approach cannot satisfy the need for accessible content (i.e. for cognitive disabled people). The authors' managers need to train authors to use appropriate / simple language for their audience / target group.

2.2 Content as aggregated portlets/subsystems

Portal systems consist of portlet applications that can provide completely different functionalities and serve completely different aims. Portlet applications consist of a number of portlets that have a common aim but at the same time they are reusable and autonomous components. Portlet applications can even be situated on different servers or different web domains WSRP and consist of fragments of third party resources. When evaluating a portal using an existing automated tool like [Bobby], the tool evaluates the produced aggregated content of a URI[4] instance. The resultant, global, report is not very useful for the designer of the portal, since the accessibility, or even validity, problems it picks up might refer to either a specific portlet accessibility problem or to an aggregation/relation problem of "web objects".

This is important as it affects the designer/administrator decisions. For instance, if the problematic portlet is a content extracted from a third party resource over which the designer/administrator has no control, the decision could be to remove the portlet or just provide a URL warning users about accessibility problems. In this case, this means that accessibility of WSRP is called into question, which leads to a requirement that user agents need to be able to extract metadata information about the source of portlets on an instance of a portal page.

In other words, portal systems need a mechanism to communicate the portal page structure to the user agents. Currently there is a portlet description language (portlet.xml) which describes a portlet application but which is used only by the portlet container and its metadata does not affect the client side. The same also happens with [PSML] a language that contains metadata about a user's page instance aggregation. It would be possible for the aforementioned information to be extracted from the combination of these languages and transferred to the user agent through headers. Then boundaries of resulting portlets could then be extracted using markup (i.e. <div id="portlet-<portlet name>-<portlet instance>"). Although the final argument could by itself satisfy the requirements of an accessibility tool in the sense of providing the name of the problematic portlet, it is noteworthy that metadata could offer better help in terms of automatic evaluation and repair features. More specifically, [JUNIT] and [EARL] could be used and extended to provide test and feedback for portlet instances.

A further important issue with content aggregation is with the so called "controllers". The controllers are actually the markup mechanism that relates the portlets with each other. Currently controllers are basically markup tables that create the "windows" for the portlets. According to web accessibility guidelines, tables for layout and moreover

[4] Uniform Resource Identifier: http://www.w3c.org/Addressing/

multiple nested tables are not allowed. That layout needs to be undertaken by CSS. Controllers also contain navigation, whose importance and complexity is discussed separately in the next section.

2.3 Complex internal navigation

For working on the navigation aspect of accessible portal systems, a good test case is made by investigating the case of mobile portal systems [Godwin & Haenel, 2002]. Mobile devices can be a very good simulation application for the designers of web portals. This is because they simulate different access problems due to their limited screen size and input capabilities that many categories of people may meet and at the same time give the marketing push to the portal customers and consequently their vendors to design with accessibility in mind.

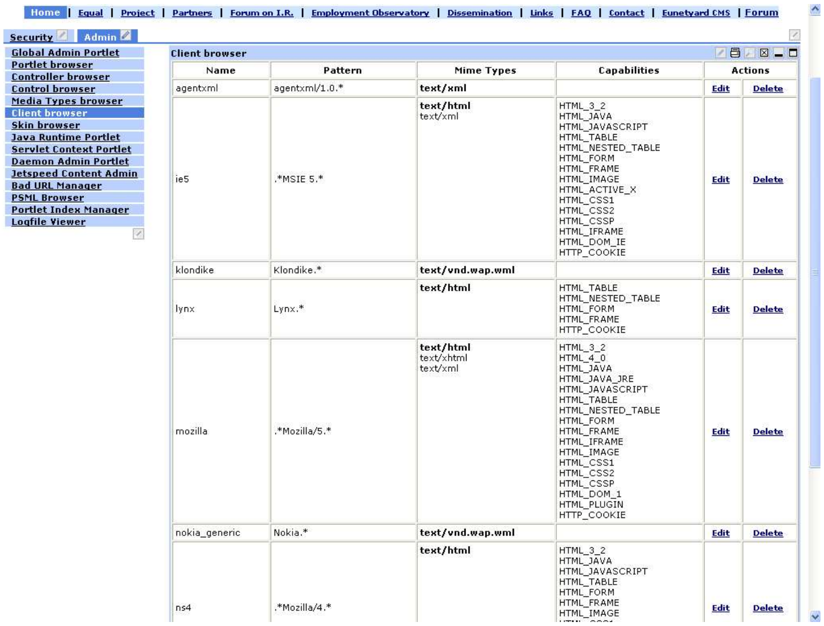
For such portals, navigation is the key to success. The users need to access the information they are looking for with not too much of navigation overhead. Artificial techniques have been used [Smyth & Cotter, 2003] for reducing click-distance and providing successful navigation. Similarly, generally speaking, in accessible web portals, users need to interact with portlets in a "portal transparent" way and also have at their disposal powerful and useful search engines: the more the information, the more the complexity of its organisation and the navigation complexity.

In a portal system, users meet two kinds of navigation: the portal and the portlet navigation (ignoring the in-portlet navigation): i.e. the users have to navigate through the portal and also through the instance of a portal page. For such a page to be accessible, it is important to make sure that when the page is serialised by an actor device it will produce an acceptable result. In other words if the serialised version of the common view of portals with the arrays of those windows / portlets is usable then it is possible to claim that this is an accessible portal page instance.

A quick solution can be provided from the aforementioned portal structure metadata. The portal system can automatically generate a portlet navigation section that can be visually transparent and also provide metadata like portlet registry descriptions. However, this is a "quick fix" and by no means is it the best solution. An alternative approach that could not only apply to portals -although these are the ones that need it- is a separation of navigation from the content.

Currently, there is much effort being put into separating the concerns (structure, presentation, logic) W3C, [Apache]. Expert groups [Cocoon] are developing methods and technologies for separating the presentation from the content in order to leave the content 'pure'. This makes possible the reuse of the content information, and more importantly for the case of accessibility, the ability to change at least the presentation of the content, so that there is the possibility to adapt it to the actor [Velasco et al, 2004]. Further, looking at the emerging corpus of work on the semantic web, the adaptivity also becomes possible at the level of the content. By separating the concerns it has been possible to separate also the designers and developers job and make it more efficient.

At the same time the navigation mechanisms become more and more complex, and run the risk of disorientating the information designer and causing accessibility problems. Navigation causes problems as it is aggregated into the content. Of course there are cases where this might be inseparable as navigation can be indissolubly linked to the content. For other cases, where also the navigation or a part of navigation (i.e. the main navigation) becomes really complex; such as in the case of portals, it is suggested that, the navigation should not form a part of the content of a web resource, but a control mechanism for viewing this resource. In other words, navigation should be metadata and not data. Thus a separation of the control of a web resource from the resource itself is needed. This will allow the actors to semantically extract the navigation information and use it to guide themselves in the web resource. The metadata can now be more that simple hyperlinks. It can also contain more information about the structure and the content of the web resource (i.e. pages descriptions or pages relations). Furthermore, this approach could offer much towards the semantic web. The navigation would then be much more easily be adapted both to the actor, or to the purpose of the content (e.g. learning) Consider for instance the scenarios that aim to be navigation guides. When users visit web page for first time, they might like to explore all web resources in a logical order depending on the web resource domain. When they come back next time they do not need to waste time navigating but find their information directly. At the same time the proposed approach offers to the actors a means of global navigation by allowing them build their personalised navigation according to their cognitive and presentation requirements / interaction capabilities. Figure 3 illustrates the portal's support for user agents capabilities that can be used for guiding decisions about presentation requirements and interaction capabilities.



Client browser

Name	Pattern	Mime Types	Capabilities	Actions	
agentxml	agentxml/1.0.*	**text/xml**		Edit	Delete
ie5	.*MSIE 5.*	**text/html** text/xml	HTML_3_2 HTML_JAVA HTML_JAVASCRIPT HTML_TABLE HTML_NESTED_TABLE HTML_FORM HTML_FRAME HTML_IMAGE HTML_ACTIVE_X HTML_CSS1 HTML_CSS2 HTML_CSSP HTML_IFRAME HTML_DOM_IE HTTP_COOKIE	Edit	Delete
klondike	Klondike.*	**text/vnd.wap.wml**		Edit	Delete
lynx	Lynx.*	**text/html**	HTML_TABLE HTML_NESTED_TABLE HTML_FORM HTML_FRAME HTTP_COOKIE	Edit	Delete
mozilla	.*Mozilla/5.*	**text/html** text/xhtml text/xml	HTML_3_2 HTML_4_0 HTML_JAVA HTML_JAVA_JRE HTML_JAVASCRIPT HTML_TABLE HTML_NESTED_TABLE HTML_FORM HTML_FRAME HTML_IFRAME HTML_IMAGE HTML_CSS1 HTML_CSS2 HTML_CSSP HTML_DOM_1 HTML_PLUGIN HTTP_COOKIE	Edit	Delete
nokia_generic	Nokia.*	**text/vnd.wap.wml**		Edit	Delete
ns4	.*Mozilla/4.*	**text/html**	HTML_3_2 HTML_JAVA HTML_JAVASCRIPT HTML_TABLE HTML_FORM HTML_FRAME HTML_IMAGE	Edit	Delete

Figure 3: Portals offer user agent's capabilities

In this paper, a navigation abstraction mechanism in a form of attached metadata (e.g. [RDF]) is proposed. This requires user agent support (for instance, in a form of plug-in or even an applet) to extract a personalised navigation, perhaps according to the user's and device profile [Velasco et al, 2003] and dynamically create navigation according to the user's preferences and/or, capabilities which will also be modality dependent. For example, one might consider a portal that can be accessed via a radio set while driving, tuned through the navigation and then making use of screen reader for listening the content or even animating on a screen. Present-day adaptive navigation techniques [Brusilovsky, 2004] could be used such as link hiding or link generation. In future this might allow "virtual portals" (aggregation of web resources in a form of navigation from a set of web resources sites) and much more. A small view of this could be seen through [RSS],which can automatically syndicate information from resources of interest and thereby create a new resource.

3 Conclusions and ways forward

This paper makes a comparison between a portal system and a web site in terms of accessibility. The additional requirements of a portal system are indicated and complementary specifications and guidelines are proposed at the design as well as at the implementation level, and suggestions are made for some types of automatic accessibility checking for portals. It should be noted that the proposed approach cannot offer fully accessible portals due to the following reasons, which are quite common in the domain of web accessibility. The first one is that semantics cannot be controlled yet even though the work on semantic web seems promising. The second reason is that the larger the variety of the structure of the portlets, the more difficult it becomes to control the portlets' content accessibility. On the other hand, in order to compensate for these difficulties, the application of standard mechanisms like specifications, multimodal channels, standards presentation, customisation, alternative presentation etc. could be provided. These would then be subject to the same types of automated checkers as are currently used on web sites. For future work, in general during the design phase of a portal infrastructure, object oriented methods like [Paloma et al, 2004], [Rossi & Schwabe (2002)] and [Ceri et al, 2000] and ontologies [Mauwa-Morgan & Burrell, 2004] could be adapted to the proposed principles and provide methods and tools for designing accessible web portal infrastructures. Last, but surely not least, the separation of navigation from the content in conjunction with user/device profiles seems very promising specifically in cases of complex navigation such as that of portals.

References

Apache Software foundation (2003). Apache community of open-source software projects. - http://www.apache.org

Barry Smyth & Paul Cotter (2003), Intelligent Navigation for Mobile Internet Portals. In Proceedings of 2003 AI Moves to IA: Workshop on Artificial Intelligence, Information Access, and Mobile Computing

Bellas Fernando, Daniel Fernández, Abel Muiño (2004), A flexible framework for engineering "My" Portals Proceedings of the 13th international conference on World Wide Web table of contents New York, NY, USA Pages: p. 234 - 243

BenToWeb - Benchmarking Tools and Methods for the Web. IST Specific Targeted Research Project. – http://www.bentoweb.org

Bobby - http://bobby.watchfire.com/bobby/html/en/index.jsp Cited 7/2/2005.

Ceri S., Fraternali P., Bongio A.: "Web Modeling Language (WebML): a Modeling Language for Designing Web Sites". WWW9 Conference, Amsterdam, May 2000.

Cocoon Apache Project - http://cocoon.apache.org/ Last cited 17/2/2005.

CSS – Cascading Style Sheets- http://www.w3c.org/Style/css

Díaz Paloma, Montero Susana and Ignacio Aedo, 2004, Modelling hypermedia and web applications: the Ariadne Development Method, Information Systems, In Press, Elsevier

EARL - the Evaluation And Report Language - http://www.w3.org/2001/03/earl/ . Last Cited 22/2/2005.

EQUAL - EUNETYARD: Promotion of competitiveness and insurance of occupation on shipbuilding industry (transational) – http://www.eunetyard.net

EQUAL - NEORIO2: Promotion of competitiveness and insurance of occupation on shipbuilding industry (national) – http://www.neorio2.gr

EQUAL – SYMPOLITIA: Islands' confederacy – Equivalent access – http://sympolitia.syros.aegean.gr

Godwin Rusty & Haenel Walter, (2002), Portal Aggregation for Pervasive Devices. IBM WebSphere Developer Technical Journal

Henrike Gappa & Gabriele Nordbrock, 2004, Applying Web accessibility to Internet portals. Universal Access in the Information Society, Volume 3, Number 1, March 2004, p 80 - 87

Hepper Stefan & Lamb Marshall (2004), Best practices for developing portlets using JSR 168 and WebSphere Portal V5.02

HTML 4.01 Specification. W3C Recommendation 24 December 1999. http://www.w3.org/TR/html4/ Cited 15/2/2004

IRIS: Incorporating Requirements of People with Special Needs or Impairments to Internet-based Systems and Services, IST-2000-26211 – http://www.iris-design4all.org

JCP - Java Community Process - Java Specification Requests (JSR 168) Portlet Specification – http://www.jcp.org

WSRP - OASIS Web Services for Remote Portlets - http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp. Cited 7/2/2005.

JUNIT - http://www.junit.org

Mauwa K. R.Morgan & Burrell P, (2004), Justification of the need for an ontology for accessibility requirements (Theoretic framework), Interacting with computers 16 523-555.

Peter Brusilovsky (2004), Adaptive navigation support: From adaptive hypermedia to the adaptive web and beyond. PsychNology Journal 2004, Volume 2 Number 1, 7-23

PORTAL - Presenting natiOnal Resources To Audiences Locally - http://www.fair-portal.hull.ac.uk/ . Last Cited 21/2/2005. Last Cited 22/2/2005.

PSML : Apache Jakarta Jetspeed Portlet Structure Markup Language - http://portals.apache.org/jetspeed-1/psml.html

RDF – Resource Description Framework - http://www.w3.org/RDF/ . Cited 7/2/2005.

Rossi Gustavo & Schwabe Daniel (2002) Object-Oriented Design Structures in Web Application Models. Annals of Software Engineering 13, 97–110, 2002 Kluwer Academic Publishers

RSS – RDF Site Summary Specifications- http://web.resource.org/rss/1.0. Last cited 17/2/2005.

Smith Michael Alan (2004), Portals: Towards an application framework for interoperability. Communications of the ACM, October 2004/Vol. 47, No. 10

U-Portal – Universities' Portal. http://www.uportal.org/. Last Cited 21/2/2005.

Velasco, C.A. Mohamad, Y. Gilman, A. S. Viorres, N. Vlachogiannis, E. Arnellos, A. Darzentas, J. S. (2004). The Need for User Information and its Relationship to Device Profiles- Journal of Universal Access in the Information Society, vol 3, no 1, March 2004, pp 88-95