User interface approaches for accessibility in complex World-Wide-Web applications- an example approach from the PEARL project

Martyn Cooper
Institute of Educational Technology/Knowledge Media Institute - Open University, UK

Liliana Patricia Santacruz Valencia
Department of Telematics - Carlos III University, Madrid, Spain

Alexis Donnelly
Department of Computer Science – Trinity College Dublin, Ireland

Paul Sergeant
Applied Computing Division – University of Dundee

Abstract

Increasingly the WWW is being used to host complex applications in diverse fields important for peoples work, study and leisure. Now the very complexity and multi-modality of such applications presents a particular challenge in seeking to design user interfaces to them that are accessible to people with different disabilities.

This paper introduces an approach that can be used to address this challenge, making complex applications accessible to people with differing needs and abilities. It is based on the ongoing work of the EU-IST funded project PEARL. An overview is given of this project together with an outline of the XML family of web technologies.

The main tactic suggested to address the accessibility problem is to separate the interface from the application such that different users can have different interfaces to the system depending on their preferences, needs and the computer equipment they are using. This paper suggests how this can be facilitated given current and emerging web technologies, particularly XML and XSL.

Introduction

Increasingly the WWW is being used to host complex applications in fields as diverse as scientific enquiry, education, e-commerce and entertainment. For the purposes of this paper we can define a complex application as one where a user has to receive multiple sources of information from a web server or several servers and is able to interact with the application in a variety of ways. This would normally take the form of an interactive multimedia application similar to that which has been traditionally produced for off-line delivery on CD-ROM. Now the very complexity and multi-modality of such applications presents a particular challenge in seeking to design user interfaces for them that are accessible to people with different disabilities.

This paper introduces an approach, and points to ways of implementing it, that can be used to address the problem of making accessible complex applications to people with differing needs and abilities. It is based on the ongoing work of the EU-IST funded project PEARL, which is developing a system for the remote control of teaching laboratories over the WWW. This project is developing a mainstream educational application but working to the access criteria that:

- Both the audio and visual modalities will be sufficient on their own
- All interactions with the system will be possible for single and multiple switch users and thus for any who use an alternative to the keyboard or mouse

The main tactic in the approach is to separate the interface from the application such that different users can have different interfaces to the system depending on their preferences, needs and the computer equipment they are using. This paper outlines how this can be facilitated given current and emerging web technologies, particularly XML and XSL.

Introduction to the PEARL project

As it is the context for the exemplars given in this paper a brief introduction to the PEARL project is given here. (For further information contact the first author or see [1].) The PEARL (Practical Experimentation by Accessible Remote Learning) project, running for 36 months from March 2000, funded by the European Commission's Information Society Technologies programme, is researching and developing a system to enable students to conduct real-world experiments as an extension of computer based learning (CBL) and distance learning systems. The objectives being to give high quality learning experiences, principally in science and engineering education, by bringing the teaching lab to the students; giving flexibility in terms of time, location and special needs. This will extend Internet course delivery to accommodate collaborative working in practical experimentation.

> **PEARL facts and figures:**
>
> •*A major European collaborative R&D project*
> – IST - Education and Training
> – Budget ~ € 2 million, Effort ~ 30 person-years
>
> •*Led by the Open University*
> – Partners: TCD, U. Dundee, FEUP, Zenon, OUW
>
> •*March 2000 - February 2003*
>
> ***See: http://kmi.open.ac.uk/projects/pearl/***

The project will develop and integrate:

- a collaborative working environment with accessible user interfaces
- a modular system for flexibly creating remotely controlled experiments
- software practical templates and tutor guidelines

The project will research the pedagogic impact of this approach, validating its developments in different educational contexts and subject areas, including:

- foundation level physical sciences (as part of an open & distance learning course)
- cell biology (as part of a final year undergraduate course)
- manufacturing engineering (post-graduate training)
- digital electronics (as part of undergraduate courses in design and testing)

Access for Students with Disabilities

The PEARL project also seeks to make experimental work in science and engineering accessible to students with a wide range of disabilities; in fact the project was conceived in trying to address access problems for disabled students in science and engineering subjects at university level. Disabled students are grossly underrepresented here. The reasons for this are complex and include significant factors from primary and secondary level education. However access to experimental work is cited as a key barrier in analysis of this problem both in the USA and Europe. Much work has been done over the past 20 years to make computers accessible to people with all kinds of disabilities and a high degree of success has been achieved. Thus for most people with a disability, seeking to participate in higher education, access technology for a computer, where needed, is commercially available. Hence an objective for the PEARL project is firstly to make practical work computer controlled and then to ensure that the computer software and interface design follow well-established design for accessibility principles [2, 3] and are compatible with the available access technologies.

The PEARL System

The PEARL system is to be implemented over the Internet or University Intranets. The system will be constructed around an XML based client-server architecture. It is currently an issue of discussion whether this will then use a complex client (such as Netscape or Internet Explorer) or a *"Thin Client"* i.e. dedicated software for the presentation of the interface to the users. The discussion of the issues here is beyond the scope of this paper and the choice does not affect the basic approach to the accessibility problem being outlined.

Central to the whole design concept is the modular, remotely configurable and controllable laboratory. This is an area of significant design and development effort within the project. An integrated design team with expertise that includes: educational experimentation, mechanical design, distributed systems, control, robotics, instrumentation, etc. has been brought together in the PEARL consortium to address this. The concept is that any particular lab will be constructed from a wide range of standard experimental elements and instrumentation. Each of these modules in the system will have a microprocessor-controlled interface to the busses, which connect all the modules to each other, and to the remote lab server. It is envisaged that this communications channel will in fact be separate proprietary control and instrumentation and audio/video busses.

The challenge of user interface design to meet broad access criteria

People have different needs when seeking to use a computer to facilitate any activity. Factors here extend beyond their physical and sensory capabilities and include needs arising from:

- preferred learning/working styles (activist / reflective) [4]
- left or right handedness
- gender
- the environment they are working in (e.g. office, train, while driving)
- other cognitive loads they have to deal with simultaneously
- … etc.

In seeking means to meet these different needs in the design in a computer application of the user interaction and the interface through which this is mediated the *"one size fits all"* approach is highly inappropriate although it is almost universally the one adopted. Some applications (e.g. Microsoft Office) give the user ability to modify the interface and save user preferences but few present the user a choice of different interfaces based on an analysis of the diversity of human need. They are basically all developed with someone similar to the software engineer actually undertaking the development work in mind as the target user (i.e. white, middle class male aged 25-35 with extensive experience of IT). The authors recommend the application developers provide different interface designs for different users - one design for all fails to meet many peoples needs and almost certainly disadvantages many disabled computer users.

How then can flexibility in a user interface to an application be achieved within the real world of financial and technical constraints and the pressure to get to market as soon as possible?

Overview of a flexible user interface approach

The main tactic

The core of the approach being put forward here is the need to separate the interface from the application such that different users can have different interfaces to the system depending on their preferences, needs, computer equipment (including any Assistive Technology) they are using, etc. Now the separation of interface and application is not novel. It is essentially what any GUI operating system does and is behind the now well-established approach in software engineering to use Application Programme Interfaces (APIs) between the core application modules and those responsible for presentation to and interaction with the user. However extending the approach to include different user interface designs, while now technically trivial, has been seldom implemented to date.

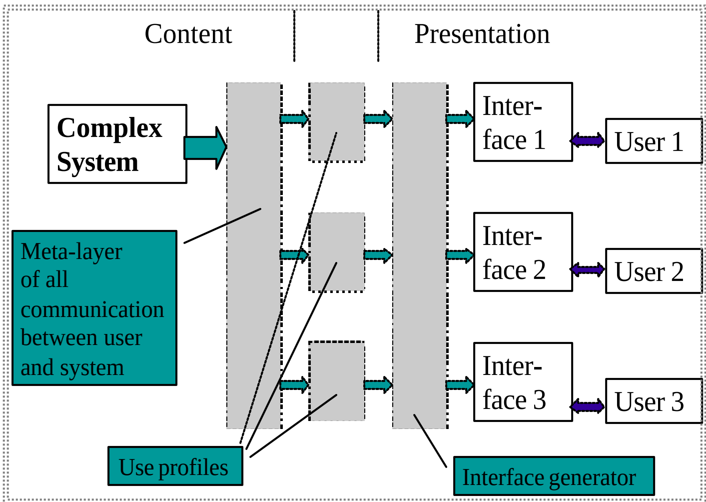


Figure 1 Simplified Schematic for Automatic Generation of User Interfaces according to User Needs

A meta-layer is defined that encompasses all the communication between the system and the users. User profiles are stored and how this meta-layer is translated to an actual interface by the interface generator is determined by these profiles. In practice what will happen in most cases is that one of a limited number of standard interfaces will be selected and then some parameters of the selected one adjusted according to the stored profile or a direct request from the user. This approach can be readily facilitated for Internet based applications given the current and emerging web technologies, in particular extensible Mark-up Language (XML) and XML based Stylesheet Language (XSL). A simplified schematic of the approach is given in **Figure 1**.

Example from PEARL project

Remote control of an electron microscope
One of the experimental facilities being implemented within the PEARL project is to enable student groups working collaboratively over the Internet to have control over an electron microscope. Communication between the user and the system is summarised in the table below.

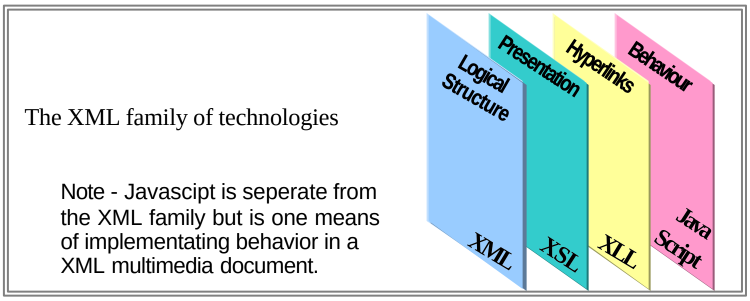
Information to user	Control from user
View of image from electron microscope Communication with fellow students/tutors Measurement data Lab script and related text/MM information	Movement of sample Control of microscope Interaction with the experiment e.g. introduction of chemical stimulus to cells Communication with fellow students/tutors

Most of these elements can be presented to the users in different ways and this is where much of the variation in possible interface design arises. For example the communications between the collaborating students could be implemented as text, voice or video link with audio. The technology exists today where by it would be possible to set up such a communications tool with different users using different modalities as their prime means of communication. Another illustration is measurement data e.g. temperature, sent back fom the remote lab. This could be represented graphically, numerically, as a background colour, textually and be voiced. Which each user prefers is likely to vary but by taking the above outlined approach these preferences could be accommodated. The above facility also illustrates that we are likely to always come up against limits to accessibility. The whole purpose of the experimental apparatus is to enable the students to observe what is happening under the electron microscope. This is obviously impossible for blind students but sighted students can describe what they see for their visually impaired colleagues – a social solution as opposed to a technical one, although in the PEARL case the social solution is mediated by the technology.

A Brief Note on XML and its related technologies

XML (Extensible Mark-up Language) is meta-language used to define other languages. Unlike HTML, XML has no defined set of tags. Its purpose is to describe data and its promise is to give the same kind of portability to data as Java has provided for program code. It should be noted that the term XML, as it is used, is generally taken to include the set of related technologies. Specifically:

- XSL
- XLL
- XSLT
- Xpath
- DOM



The term XML is thus used to cover whole series of related technologies that perform different roles within a complex web-based application. Key technologies and their roles are illustrated in the graphic above. These can be considered as separate layers within a multi-media "document".

XML is a "declarative" framework on which mark-up codes (other mark-up languages) for any application area can be generated. Unlike HTML, in XML logical markup (which delineates document structure) is separated from Physical markup (which describes how that logical content should be printed). The sadly common HTML practice of mixing physical and logical markup creates problems for the designers of assistive technologies (e.g. screen readers).

To appreciate the potential of XML it is important to have some understanding of these technologies, which are now outlined. Therefore for the rest of this section, the term "XML" can be taken to have the more inclusive meaning.

Purpose of XML

XML was designed to address the two principal weaknesses of HTML. Firstly HTML, as it has developed, fails to separate form from content. The limited information about what parts of the document are is mixed up with instructions about how to present it. Secondly, HTML is just too limited, e.g. it is not possible to include meaningful semantic tags. This lack of flexibility becomes more acute as information providers have to support a wider range of clients (set top boxes, WAP phones etc.)

XSL

XSL stands for Extensible Stylesheet Language. Unfortunately the inclusion of the term "stylesheet" is confusing because XSL is concerned with much more than layout. It has two purposes:

- to be a declarative language for describing transformations of XML documents.
- to be a vocabulary for specifying the format of XML documents.

The clear distinction between these two purposes has lead to the transformation task being given a new name XSLT. The formatting component is still officially named XSL, but it is often referred to as "XSL-FO" (XSL formatting objects).

XSLT

XSLT stands for Extensible Stylesheet Language: Transformations. The specification can be found at XSLT [5]. It is a declarative language of which the principal purpose is to describe XML document transformation. Its primary purpose was to effect the transformation from one XML document into another. However it can also be used to carry out a transformation in to any text-based format—a frequent use is to transform XML to HTML.

Xpath

Xpath results from a merging of the expression syntax of XLST for selecting parts of a document and the Xpointer language that was being developed for linking from one XML document to another.

DOM (Domain Object Model)

DOM is an API for converting XML documents into tree-like structures – all data is stored as nodes that can be manipulated by programming languages. Although DOM gives full access to XML data, it can be an inefficient way of processing large documents because the whole thing must be read in before any processing can be performed (this is not the case with simple parsing).

Uses of XML

XML and its associated technologies are being used for a wide variety of tasks including:

- Presentation
- Inter application communication
- Communication between processes
- Configuration

Presentation

XML technologies' role in presentation is the separation of content from presentation through XSLT (described above). It is powerful combination because XML data is easily searchable and XSTL can describe transformations for presentation in any order (something not possible with CSS – Cascading Style Sheets).

Inter application communication

XML is increasingly being used to transfer data between applications. This includes use in business-to-business (B2B) communication. XML is ideal for this because they are formally described, hierarchically structured and in human readable form. In addition, through XSLT they can be easily transformed.

Inter Process communication

The specification for XML base RPC (Remote Procedure Call) can be found at RPC [6]. RPC is used for making procedure call over the network. However it is not suitable for remote object communication. The latter is the domain of technologies such as CORBA and RMI, which communicate with object methods via proxies (stubs and skeletons). However the XML mapping between textual representation and structure means that RPC based communication is now feasible in an object-oriented environment.

Configuration

XML is increasingly being used for configuration files. For example, the EJB (Enterprise Java Bean) specification 1.1 states that XML be used for deployment descriptors and Apache TOMCAT (the reference Servlet/JSP platform also uses XML to describe parameters.

Delivery

XML pages can be delivered through a web browser in three ways:

- direct to an XML aware browser
- dynamic processing on the server
- pre-processed XML to HTML transformations on the server

Direct processing by an XML aware browser has the advantage of reducing server load. The downside is that this is beyond the capability of most browsers. Explorer 5.0 upward offer XML support, but it is an early non-standard implementation (though Microsoft has promised to support standards). For now, the best option is dynamic server side processing because this can be adapted to support a wide range of clients.

Summary

The above outlined functionality should demonstrate why XML is becoming increasingly selected to implement complex applications over the Web. But in the context of this paper the key fact is that it provides the separation of content from presentation required to implement the proposed flexible user interface approach of the PEARL project and which is being advocated more generally here. Further the XML family provides tools ready-made to implement this.

Role of XML/XSL in flexible interfaces

The role of XML and XSL in implementing the flexible interface approach is illustrated in the schematic right. Essentially a server-based application generates a dynamic XML page on the client computer. The user profile selects which XSL sheet is used to determine how the XML based content is presented to the user in question. The detail of the XSL sheet can be set by automatically loaded defaults based on:

- an assessment of the user's needs
- a store of past preferences
- modifications made directly by the end-user

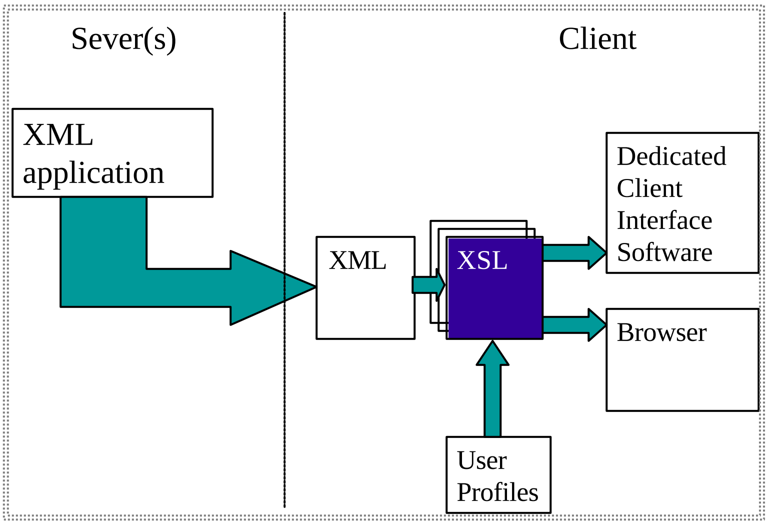


Figure 2 Schematic illustrating the use of XML and XSL in creating diverse user interfaces to a complex Web application

The extensible nature of XML means that new tags (representing for example equipment controls or gauges) can be added to formerly pure HTML web pages. XML aware servers can identify and replace these tags based on equipment state before transmitting them to a client. Likewise, XML aware client software can present new tags in a user-tailored fashion determined by user profile/preference. Further, experimental data, tagged in XML, might be transmitted to a plugin associated with the browser for later rendering in a user-tailored manner using the XSL part of XML.

In Conclusion

Complex Web based applications exist and will increasing do so and become more important in people's day-to-day lives for leisure, study and work. If disabled people are to be enabled to use these it is imperative to address the user interface issues in a flexible way, one interface will never meet the needs of all users. Web technologies now exist that enable us to implement systems with different user interfaces being automatically generated depending on user profiles. XML with its associated technologies offers great potential here and all designers and developers should think about how the inherent advantages of this technology over its predecessors can be enhanced to improve usability and accessibility for all.

References

1 **Cooper M., Scanlon E. and Freake S. L.,** Remote Controlled Teaching Experiments, in Science and Engineering Subjects, Accessible over the World-Wide-Web - The PEARL project, *Proc. ED-MEDIA 2000 Conference*, Montreal, Quebec, Canada, June 2000

2 WAI Content Guidelines. Available at: http://www.w3.org/TR/WAI-WEBCONTENT/

3 Microsoft Windows Guidelines for Accessible Software Design (1999). Available at: http://www.microsoft.com/enable/dev/guidelines/software.htm

4 Honey, P. and Mumford, A. (1986) The Manual of Learning Styles, Peter Honey

5 [XSLT] http://www.w3.org/TR/xslt/

6 [RPC] http://www.xml-rpc.com/