AN INNOVATIVE TOOL TO EASILY GET USABLE WEB SITES

Cosimo Antonio Prete, Pierfrancesco Foglia
Dipartimento di Ingegneria dell'Informazione, Facoltà di Ingegneria, Università di Pisa, Via Diotisalvi 2,
56126 Pisa (PI), Italy

Michele Zanda
Scienze ed Ingegneria dell'Informatica, IMT Alti Studi Lucca, V. San Micheletto 3, 55100 Lucca (LU), Italy

Keywords: Web usability, development tool, navigation, fast prototyping, presentation, colour.

Abstract: This paper describes the actual methodologies to develop usable web sites. We consider significant tools to model web sites and pages, and then we propose an innovative approach to create usable web sites rapidly and easily. Our goal has been the inclusion of new methodologies in the web application development process.

1 INTRODUCTION

Making a site that follows the guidelines of the web usability theory is one of the key factors to have a web site with a high goal completion rate (Nielsen, 2003). Despite the simplicity of the statement, it's hard to find a good approach to make web sites usable. The two main reasons may be: researchers and web developers have focused their attention on code generation and in a second phase on web presentation; web usability is thought to be mainly an 'art' and thus it cannot be managed by the software. The first part of this work wants show the actual tools, while in the second one we present Easy Usable Web sIte (EUWI), a new development environment for web sites.

2 PROBLEM STATEMENT

The main problem in the industrial approaches is that usability is usually faced at the end of the web development process, with high costs and growing times to the final release. The lifecycle of a web application has been well depicted in figure 1 (Fraternali, 1999), and we are showing his scheme slightly changed.

High costs and growing development times are generated by the two loops. These refinement loops are caused by different opinions and visions about the final structure and presentation of the web site between the web developer, the client and the final users (if asked).

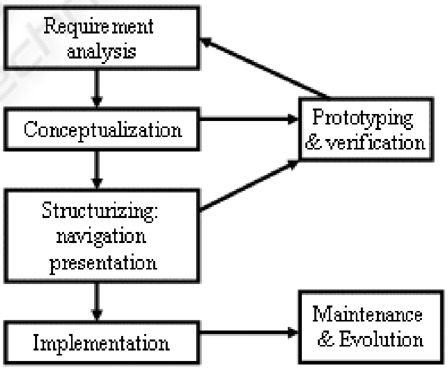


Figure 1: The lifecycle phases of a web application. It can be seen that there are two close loops that postpone the implementation phase

This paper tries to propose a methodology and a tool to avoid these loops by including usability factors at the beginning of the web development process, rather than at the end.

3 EXISTING METHODOLOGIES

At the moment there are a lot of development tools and methodologies, already released or under development.

Antonio Prete C., Foglia P. and Zanda M. (2005).
AN INNOVATIVE TOOL TO EASILY GET USABLE WEB SITES.
In *Proceedings of the First International Conference on Web Information Systems and Technologies*, pages 373-376
DOI: 10.5220/0001236503730376

Rapid Application Design and Testing (RAD-T (Becker, 2002)) is an innovative development tool mainly thought to include usability requirements inside the HTML source code. Since the traditional HTML provides little support for the usability constraints, the Berkeley researchers have created the Self Testing - HTML (ST-HTML) to extend the HTML functions. ST-HTML pages acts as a sort of active ones. The main drawback in this approach is that ST-HTML is not supported by current browser, so at the moment it can be used only as a specification tool. The research group is trying to map ST-HTML onto XML to use it with XML enabled browsers.

Web Modelling Language (WebML (Ceri, 2003) models data intensive applications in short time. Its main purpose is code generation, so as to obtain a working site, possibly even the definitive site. Later the CASE tool WebRatio has been developed to work with WebML. The web pages are rapidly structured and traversed in a GUI, and then presented by page templates or by XSL descriptions. At the end, with XML and XSL, the pages source code is generated. In the overall process, usability aspects are faced at the end, when the web developer writes or imports the presentation code.

The tool DENIM (Newman, 2000), considering the common practice of paper prototyping, especially in the initial phases of a web site development, wants to combine the paper prototyping properties with the digital qualities. DENIM consists of an electronic blackboard with pages drawn roughly and connected with arrows. The blackboard area has different zoom levels, to visualize different aspects of the site: from a general navigation structure, to storyboards, to single pages. It is also present a simple navigation facility.

To provide a better support to the communication between web designers and developers, the traditional UML Use Case diagrams have been enriched to include new features: usability requirements, pre and post conditions, controls properties, tab order, colours range ... (Hammar Cloyd, 2001) The use of WWW Use Cases can help to reduce the web applications time to market.

The need to model the interaction between web application and final user has brought Conallen to develop his User eXperience modelling (Conallen, 2003). The UX diagrams model the storyboards and the dynamic information of the web pages; by looking at the diagrams it is clear how the user will reach his goals and how wide and deep is the web structure.

4 THE EASY USABLE WEB SITE SOLUTION

Looking at the actual development tools and methodologies, we have realized that the market needed a web pages fast prototyping tool, enriched with a navigation editor and a way to naturally enforce the usability requirements.

The Easy Usable Web sIte (EUWI) tool manages to automate many areas. Here is a short list:

- storyboard depths, according to the users access frequency and goals;
- pages maximum sizes (in terms of bytes and screens), according to the type of page, to ensure a short loading time;
- presence of web objects (links to home page, backward and forward buttons,…);
- ‘alt’ text that must be included within the images;
- the elements of the menus must be between 3 and 10 units;
- font family and size to ensure good readability;
- text colour and background colour chosen to ensure good readability;
- ‘tab’ order, as to ensure that a first form field would have the focus selected automatically;
- status bar notifications.

The EUWI tool has been developed under the framework of the Model View Controller (MVC), so the web site can be seen with 5 different views: the navigation editor where the pages are created and linked, the fast prototyping environment, the ‘style and colour’ chooser, the source code window and a simple browser preview window.

The navigation editor (figure 2) is thought as a

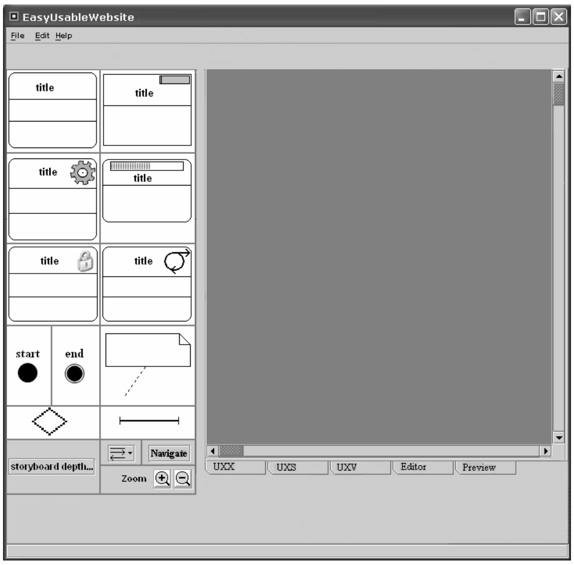


Figure 2: The UXX Navigation Editor. In this environment the web developer can depict the navigation structure of the site. Once the site pages are dragged in the central area, they are linked according to the required paths. The diagrams offer a navigation facility, and a table to choose the right wizard length.

'drag&drop' environment based on the User eXperience eXtended model.

The main UXX features are an easyreading layout (with class stereotypes being represented by different frames and start and end icons) and strong form data typing to be easily integrated in a development environment. The data types that have been codified for the forms fields are: String, Password, Text, Number, Date, Time, CCN (Credit Card Number), Email.

Multiple elements are handled with the suffix []. Further input specifications can be included with three keywords: combobox, checkbox, radiobutton.

The UXX navigation editor is used to draw navigation diagrams. The navigation diagrams (figure 3) show the site depth and the storyboards; these diagrams can be then presented to the web developer client to give him a glance of the site structure.

Once the pages are created and described in their dynamic content, they can be completed by setting the static information with the fast prototyping tool.

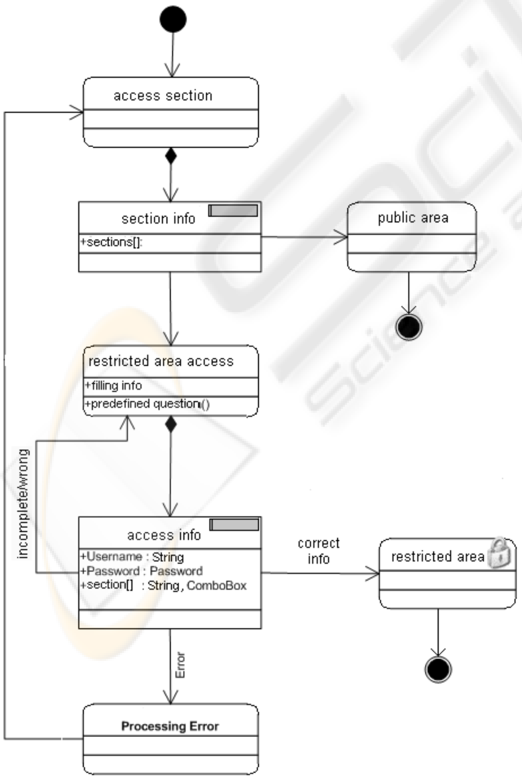
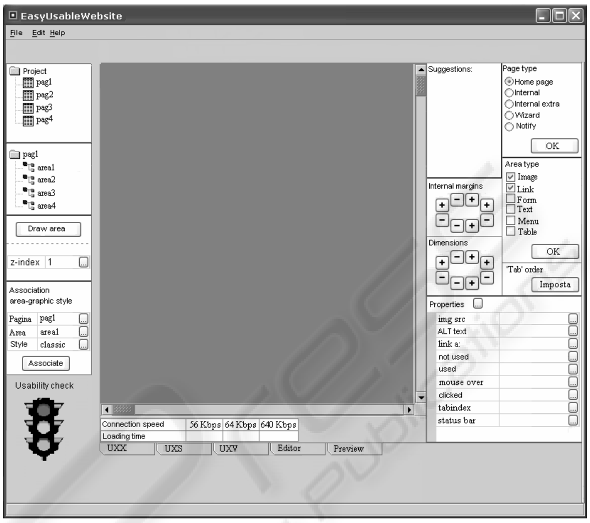


Figure 3: A navigation diagram with the User eXperience eXtended (UXX) model. The access pages are composed by input forms. If the Internet user inserts correctly login and password, he can enter the restricted area with a secure connection. The diagram has one entry point and two exit points.



z-index 1

Association area-graphic style

Pagina pag1
Area area1
Style classic

Page type
● Home page
○ Internal
○ Internal extra
○ Wizard
○ Notify

Area type
☑ Image
☑ Link
☐ Form
☐ Text
☐ Menu
☐ Table

'Tab' order

Properties
img src
ALT text
link a:
not used
used
mouse over
clicked
tabindex
status bar

Connection speed	56 Kbps	64 Kbps	640 Kbps
Loading time			

Figure 4: The Fast Prototyper environment. As shown by the above screenshot, the web developer can choose (in the top left corner) the page to describe among the pages inserted in the UXX navigation diagrams. The developer can draw rectangles in the central area to prototype the static (with its presentation) structure of the pages. The rectangles can be chosen with a check box from: image, link, text, form, menu, table. The type of page can be chosen between: Home page, Internal, Internal extra, Wizard and Notify. In the bottom left corner there’s a traffic light for usability warnings.

The fast prototyping tool (figure 4) is thought as to provide a central window to define the page working areas. Then the set areas can be specified by choosing their types (checkbox choice at the top right), their static information and relative attributes. The attributes that can be set are suggested to follow usability requirements.

For example, if the developer draws an ‘image and link’ area, he will be able to set the relative properties: image source file, alternative text, CSS link selectors (link, visited, hover, active), tabindex and status bar text.

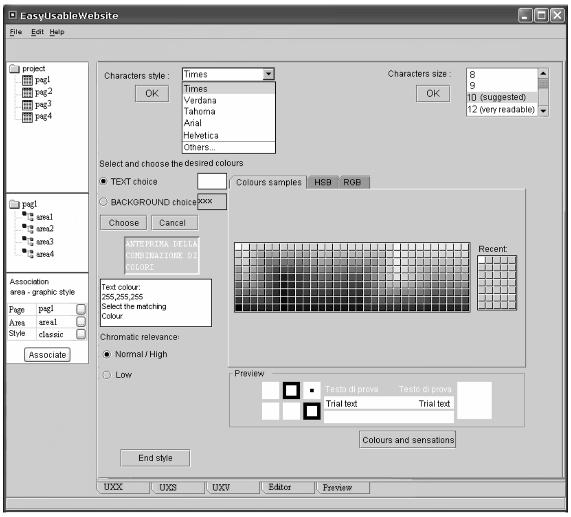
The page structures can be saved and used later as templates for new pages to accelerate the development process.

Finally, it is foreseen an interface (an addon to the former one) to set the colours of text and background so as to ensure a readable text, and to communicate the chosen sensations (figure 5). The colours constraints have been taken from the W3C

(W3C, 2004) and included in the GUI based software.

The styles of the text characters are yet suggested to satisfy usability guidelines: Times, Verdana, Tahoma, Arial, Helvetica.

To ensure perfect readability, even the size of the characters can be chosen among a minimum of 8 pts, 10 pts (preferable), 12 pts for people with visual deficiencies or bigger ones. Since at the moment the human computer interaction is not as complete as in real world, every web object must be carefully chosen to make the final user feel fine. So it is important to choose colours correctly to communicate the right sensations. Our software helps the web developer by giving him the correspondence between every colour and its meaning in various cultures (western, oriental, ...). That is very useful if the web site is not a world site, but a regional site or a site for a specific group of users.



Characters style : Times

Characters size : 10 (suggested)

Select and choose the desired colours

● TEXT choice

○ BACKGROUND choice xxx

ANTEPRIMA DELLA COMBINAZIONE DI COLORI

Text colour:
255,255,255
Select the matching Colour

Chromatic relevance:
● Normal / High
○ Low

Association
area - graphic style

Page	pag1
Area	area1
Style	classic

Figure 5: The fast style prototyper. In this environment the web developer can choose some presentation aspects about the page areas drawn before. The developer can set text and background colours to ensure a good reading. The colours can also be chosen according to the sensations that the site must transmit, depending on the culture of the typical Internet user. On the left the created styles can be associated with the page areas drawn in the prototyping tool.

5 CONCLUSIONS

The idea to include the colours constraint for text and background proves to be valuable, since, according to a web analysis, at least the 33% of the observed pages doesn't satisfy the colours requirements. Forcing (or warning) the web developer to follow usability guidelines, produces better web presentations, and shorter release times. However it must be kept in mind that the presence of usability professionals might still be a necessity, as in complex cases the software can only help the human decisions.

The code generation has been thought to be based on HTML plus CSS, since the CSS and the box model are an easy way to generate the page areas drown in the fast prototyping environment.

The tool under evaluation, and the methodologies illustrated, will improve the Internet users satisfaction and will shorten the web sites time-to-market with advantages for developers and clients.

The EUWI CASE tool still needs to be properly tested for a final release. The usability methodologies (the ones that could be automated) have been identified, specified and included in our tool.

REFERENCES

Nielsen, J., 2003. *Designing Web Usability: The practice of simplicity*, New Riders Press.

Fraternali, P., 1999. Tools and Approaches for developing Data-Intensive Web Applications: A Survey, *ACM Computing Surveys*, Vol. 31, No. 3, September 1999.

Becker, S. A., Berkemeyer, A., 2002. Rapid Application Design and Testing of Web Usability, *Multimedia IEEE*, Vol.9, Issue:4, Oct.-Dec. 2002, Pages: 38-46.

Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., Matera, M., 2003. *Progettazione di dati e applicazioni per il web*, McGraw-Hill, Milano. Internet: http://www.webml.org
Internet: http://www.webratio.com

Newman, M. W., Landay, J. A., 2000. Sitemaps, Storyboards, and Specifications: A Sketch of Web Site Design Practice, Symposium on Designing Interactive Systems, Proceedings of the *Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, New York City, United States, Pages:263-274, ACM Press.

Hammar Cloyd, M., 2001. Designing User-Centered Web Applications in Web Time, *Software IEEE*, Vol.18, Issue:1, Jan.-Feb. 2001, Pages:62-69.

Conallen, J.,2003. *Building Web Applications with UML*, Addison-Wesley Object Technology Series, Boston.

W3C, 2004. *Techniques for Accessibility Evaluation and Repair Tools*. (n.d.). Retrieved November 8, 2004, from: http://www.w3.org/TR/AERT