

An Accessible Platform for Conference Administration and Management

George Margetis¹, Stavroula Ntoa¹, Maria Bouhli¹, and Constantine Stephanidis^{1,2}

¹ Foundation for Research and Technology – Hellas (FORTH),
Institute of Computer Science, Heraklion, GR-70013, Greece

² University of Crete, Department of Computer Science
{gmarget, stant, bouhli, cs}@ics.forth.gr

Abstract. Nowadays online conference management systems are an essential tool for the organization of a conference. However, most such systems do not provide support for accessibility by disabled users. This paper discusses the design and development of a platform supporting the development of accessible conference management systems, focusing on accessibility integration through the use of an iterative approach. Accessibility issues as they were raised during the evaluation procedure are also discussed.

Keywords: Accessibility, model-based web application development, conference management.

1 Introduction

Web-based systems for the management and administration of conferences have evolved into sophisticated tools for addressing the needs of conference participants and organizers, providing a large number of functionalities to support conference management tasks. There is a plethora of such systems available, commercial or open-source, supporting a variety of tasks, such as electronic submission, review, advocacy, and acceptance of papers for the organizing committee, or paper submission, editing and user profile facilities for simple users. However, support for accessibility is lacking or minimal in most cases.

On the other hand, the need for web resources to be accessible to people with disabilities has become compelling. A number of guidelines have been published by the World Wide Web Consortium, the European Commission, national governments and others during the last years in order to address the issue of web accessibility. The most well-known guidelines for web content developers are the Web Content Accessibility Guidelines (WCAG) [1].

However, creating an accessible web service is not just a matter of applying these guidelines. Instead, a more complex procedure is needed in order to ensure the desired level of accessibility, engaging web designers, web accessibility experts, web developers and of course representative end users. In this paper, the design and development of an accessible platform for conference administration and management is discussed, and the iterative procedure that was applied, as well as some of the

interesting accessibility issues that occurred, are presented. Accessibility evaluation was placed at the center of the iterative development procedure, resulting in minor redevelopments through the development lifecycle. Last but not least, it should be mentioned that the platform was designed to support not only the development of accessible conference management systems for users with disabilities, but also of system usable for everyone.

2 Background

Modern web based applications follow the Model-View-Controller design, where Model represents the application business logic or state (e.g., the data stored), while View represents the User Interface elements of the application and Controller represents the navigational code. In Figure 1, the lifecycle of a Model-View-Controller design is represented, where the solid lines indicate a direct association and the dashed lines an indirect one. In more detail, the Controller is responsible for handling events (e.g., user actions) and updating the Model accordingly. Control is usually then forwarded back through the Controller to the appropriate View.

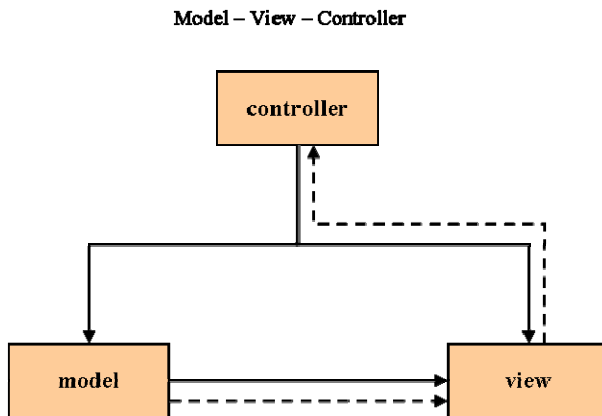


Fig. 1. Model-View-Controller design representation

The main advantage of this approach is that it allows the separation of the semantic information regarding the use of objects, their relations and meanings from the way this information is displayed to the users, providing thus the ability to isolate the user interface from the business logic of an application. Therefore, changes to the user interface do not impact the data handling, while data can be reorganized without changing the user interface. In modern web applications, data in the View part is displayed through a combination of templates, facilitating thus accessibility integration.

In order to facilitate accessibility in a database driven application, quality control on inserted data and to user interface must be ensured [2]. Quality control concerns the validation of new content for conformance with accessibility guidelines as a part

of the web application. Moreover, the evaluation process should include additional tests for the accessibility of content which is formulated as a result of the combination of templates encapsulating accessible content chunks. Interface accessibility refers to the employment of W3C Web Content Accessibility Guidelines to ensure the accessibility of form and navigation elements the user interacts with. In web applications that use templates as common containers for data representation, accessibility integration mainly results from accessible templates design. If templates are well-designed and the content creation process is strictly controlled for accessibility, the web application is likely to meet accessibility standards [3].

The purpose of the developed platform was to provide the necessary infrastructure in order to enable the development of accessible conference management applications. For the design and development of the platform, an open-source Content Management Platform [4] was used and appropriately expanded to incorporate all the functional modules needed to support the organization and administration of a conference. The open-source platform that was used enables the development of web applications following the Model-View-Controller design, which is suitable for accessibility integration. Moreover, it provides core functions for database communication, dynamic web content presentation with the use of templates and sub-templates placeholders, as well as users and permissions management.

Regarding the development of the accessible platform, quality control issues were negligible due to the nature of the content. More precisely, the content that is inserted in a Conference Management System mostly consists of unstructured plain text (e.g., paper title, author names, email addresses, dates, numbers, etc.) that can be easily encoded into a strict semantic scheme, such as XHTML, which is the necessary precondition for conformance with accessibility guidelines. Therefore, efforts towards accessibility integration mainly focused on user interaction and navigation issues, which will be further analyzed in the next sections.

3 Conference Management Platform Description

Besides support for accessibility, the main goal in the design and development of the conference management platform was to provide a variety of functionalities, in order to address the needs of both the conference organizing committee and of conference participants.

Functionality provided to conference participants includes:

- Paper submissions, in three stages: abstract submission, paper submission and submission of camera-ready version of the paper
- Tutorial and poster submissions
- Conference Registration
- User profile management.

Functionality provided to the conference organizing committee includes:

- Reviewing process administration, encompassing the assignment of papers to a number of reviewers, reviews administration and authors notification
- Program organization, including advance program and final program editing

- Submissions administration, allowing submissions categorization by thematic area as well as the administration of authors submissions
- Users administration
- Student volunteers administration
- Administration of participants' conference registrations.

4 Methodology

The process that was applied for the design and development of the conference management platform was iterative, mainly focusing on accessibility evaluation. Following the Model-View-Controller approach, the design of the conference management platform focused on creating accessible XHTML templates for every functional module. Therefore, the platform ensures that conference management systems based on it will have all the required functionality and will be inherently accessible. Conference management system developers will only have to adapt the look and feel of their application (e.g., style sheets, images, etc.).

The platform development process, as shown in Figure 2, was initiated with the Requirements Elicitation phase, the results of which were used for the design phase.

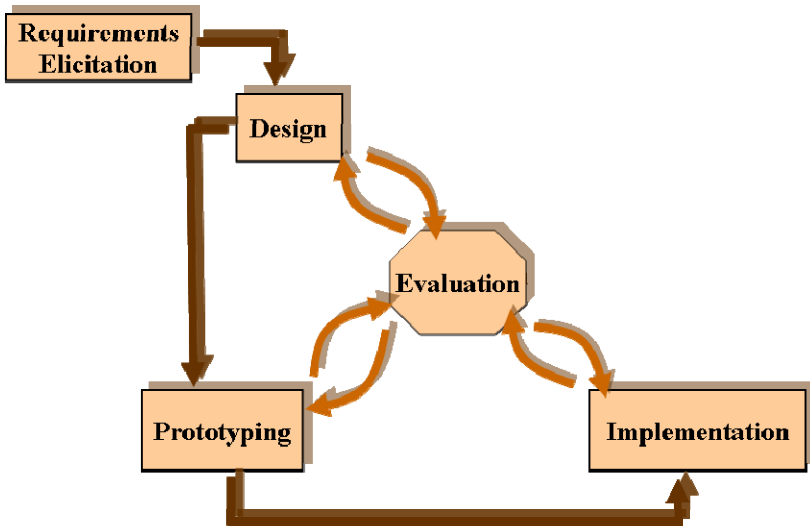


Fig. 2. Iterative Design Process

The design process included the creation of mock-ups and their evaluation by accessibility experts, using detailed checklists of accessibility features that should be later included during the development of the platform, based on W3C WCAG 1.0.

Once the design of the platform was finalized, interactive prototypes were created in XHTML. These prototypes were evaluated for their accessibility with the use of an automated tool [5].

Finally, after having ensured the accessibility of the prototypes, the implementation phase, based on these prototypes, was initiated. Implementation was alternated with accessibility evaluation until the desired level of accessibility was reached. A more detailed discussion regarding the evaluation of the implemented conference management platform is addressed in the section “Accessibility Evaluation”.

After the prototyping phase, all the individual accessible templates that were required in order to compose the overall user interface were created. In particular, a main container template was used to structure the main user interface layout by defining five basic areas of functionality: header, navigation menu, navigation path, main page content and footer.

Header includes the application logo and the quick access links. Navigation menu comprises the main functionality provided, organized in a two-level menu. Menu options are differentiated according to the user role (e.g., conference participant, author, reviewer, administrator, etc.). Navigation path consists of a number of links of increasing granularity, according to the user selections. Footer displays information regarding the compliance of the application to the accessibility guidelines. Finally, the page content template can be further decomposed to a combination of containers of one or more templates, according to the specific system functionality. The position and style of the containers in which the sub-templates are displayed are specified by the page content template handler.

The screenshot shows a web browser window displaying the 'FORTH-IES Conference Management System'. The page is annotated with five areas:

- Header:** Includes the system logo and title 'Conference Management System'.
- Navigation menu:** A vertical green sidebar on the left containing links: Home, Submissions, User Profile, Registration, Student Volunteer application, and Logout.
- Navigation path:** A horizontal bar above the main content area showing 'User: test@user.com' and 'You have selected (Home)'.
- Main page content:** The central area containing:
 - 'Submitted proposals' section.
 - 'Proposed invited papers' section with a list item: '• 3250 Test paper title'.
 - 'Registration' section with a table:

Date ordered	Order ID	Order Type	Status
Apr 01, 2007	IICI2007-1175419390336903	Remittance - Wire Transfer	Pending
- Footer:** Includes logos for W3C, WSC, and AAA, along with a small text '1.852s'.

Fig. 3. Summary page of a user account explaining the use of five basic areas of a potential Conference management system

For example, Figure 3 illustrates the summary page of a user’s account of a testbed conference management system, displaying information regarding submissions and registration, each constituting a different sub-template derived from a separate model handler.

5 Accessibility Evaluation

Accessibility evaluation during the implementation phase was carried out as a combination of manual evaluation and evaluation with the use of tools. A representative sample of pages was selected, including pages with a variety of layouts and functionality.

Manual evaluation of the selected sample included their examination with different graphical web browsers. The most popular web browsers were used (Internet Explorer, Mozilla Firefox, Opera) in their latest and one older version. For each browser, the following adjustments and checks were performed:

- Images were turned off and the availability and appropriateness of alternative text was examined
- Browser controls were used to change the font size, while page layout was checked to determine whether web pages were still readable at larger font sizes
- Different screen resolutions were tested and the page layout was checked to verify the absence of horizontal scrolling
- Color was changed to gray scale in order to determine whether the color contrast was adequate
- Keyboard was used to navigate through the page content, ensuring that access to all interactive elements was provided, in an appropriate order
- Scripts and style sheets were disabled and pages were tested to verify whether the functionality was still available and the layout was usable.

Furthermore, the selected pages were examined with the use of a text browser [6] in order to ensure that all the information and functionality were provided and that information was presented in a meaningful order.

In addition, the following automatic and semi-automatic evaluation tools were used:

- W3C Markup Validation Service [7], in order to check XHTML conformance to W3C Recommendations
- W3C CSS Validation Service [8], in order to check the style sheets that were used
- Watchfire® Bobby™ [5], in order to examine the pages' conformance with the W3C WCAG.

It should be mentioned that pages were also evaluated regarding the colors that were used, in order to verify whether the brightness and the contrast between the text and the background colors was adequate. To this end, two on-line free tools were employed. The first was AccessColor [9], a report generating tool, according to which both color difference and color brightness meet the recommended standards. The second tool was ColorFilter [10], simulating the look and feel of a page for various user categories with vision impairments, such as protanopia, deuteranopia, etc., according to which all the displayed information is readable by all the simulated user categories.

The result of these evaluations lead to some development-evaluation iterations, until the desired level of accessibility was reached. It should be noted that iterations were minor and the accessibility problems that were reported were few, mainly due to the accessibility evaluation procedures that were applied during the design and

prototyping phase. In more detail, during the design phase, accessibility experts provided guidance regarding major accessibility problems that should be avoided. Evaluation in the prototyping phase ensured the accessibility of each individual sub-template that would be used for the composition of the main interface. Therefore, accessibility evaluation of the implemented interface mainly focused on issues regarding the combination of the sub-templates as an overall accessible interface.

6 Accessibility Issues

During the iterative process of the design and development of the developed conference management platform, several accessibility issues were identified. Besides the well-known ones, such as the necessity of a descriptive alternative text for image, or the use of style sheets for the layout of a web page instead of layout tables, some other interesting concerns also came up.

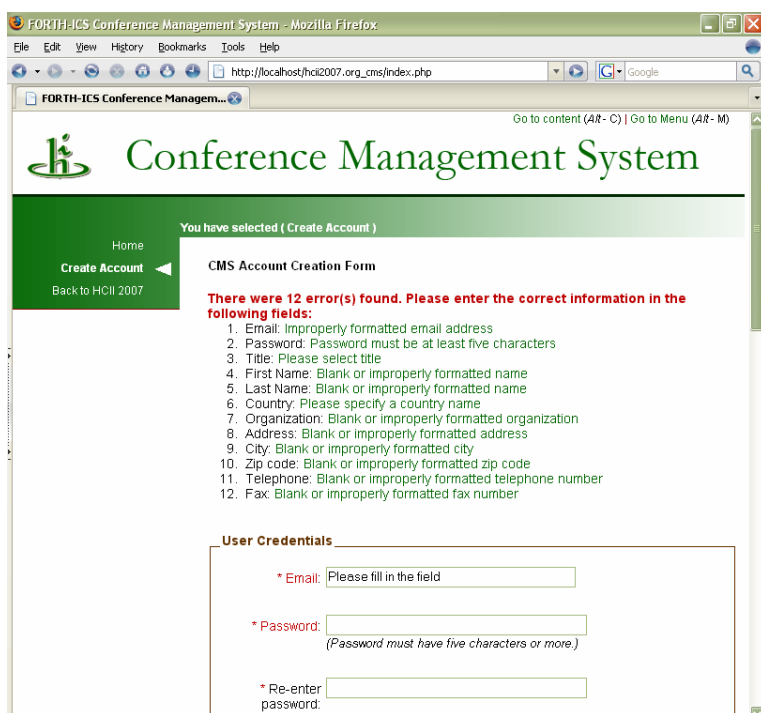


Fig. 4. Accessible error handling in forms

An interesting issue is that of error handling in forms. Usability experts usually suggest placing error messages in red color below the erroneously completed fields and to mark these fields with a red color as well. However, this approach is not usable by people with disabilities. In more details, the main drawbacks are the use of color to

signify information, which violates accessibility guidelines, and the increased effort required by blind users in order to locate the errors.

An alternative approach, as shown in Figure 4, is to place all the error messages to the top of the content, allowing direct access to error messages for users with vision impairments. Furthermore, error descriptions can be linked to the related field labels, enabling thus immediate interaction with these fields both for people with vision impairments as well as for people with motor-impairments of the upper limbs. Therefore, effective and efficient interaction is ensured in an accessible way.

Another issue worth mentioning is that of navigation. It should be possible for users of assistive technologies (e.g., screen readers) to easily and efficiently access the main page areas (e.g., menu bar, page content). This can be achieved with the use of quick access links on the one hand and skip links on the other. Quick access links are the topmost elements in every web page, providing fast access to main areas of the page and therefore the flexibility to directly move to a point of interest, avoiding interaction with preceding elements. For example, in the case of a blind user with a screen reader, interaction with the main page content is possible without the need of repeating already known information, such as the description of the site logo or the navigation path, on every new page request. Skip links in conjunction with quick access links further enhance user interaction, providing the ability to skip specific areas of information. These links are only visible when style sheets are disabled and page content is serialized.

An additional accessibility problem that occurred during the development was related to radio buttons that were initially selected to be used in forms. Some of the radio buttons accessibility drawbacks are:

- They do not scale along with the text on the page, making it difficult for users with low vision to determine whether they are selected and impeding users with motion impairments from selecting them because of the small target.
- Since in a radio buttons group many form controls are involved, explicitly associating the prompting text with form controls is impossible, misinforming thus users with screen readers.
- They introduce multiple form controls into the tab order, resulting in extra effort for screen reader users to navigate a form.

The alternative solution that was preferred was the use of selection boxes, which address all the aforementioned issues. In more detail:

- they scale along with the text
- the prompting text can be easily and accessibly associated to the select element since there is only one form control involved,
- due to the uniqueness of the form control, tab navigation is feasible.

Finally, the issue of handling deep link users [12] is also worth mentioning. Using bookmarks is convenient for people with disabilities, in order to directly access an often visited web page (e.g., information regarding an author's submissions review results). The developed platform supports stateless requests for most of the functionalities provided (i.e., users do not need to follow a specific interaction path), with the only precondition that user's browser has stored a unique identification cookie for this user, in order for the system to automatically log the user in.

7 Conclusions and Future Work

This paper has discussed the design and development of an accessible platform for conference management and administration. It was developed based on an open source Content Management Platform, enhancing accessibility with the use of templates. The process that was followed for the design and the development of the conference management platform was iterative, alternating the design, the prototyping and the development phase with accessibility evaluations. This approach resulted in minor redevelopments after each evaluation phase, reducing therefore the resources needed for accessibility integration.

This paper has discussed the main accessibility issues that were identified during evaluation, including error handling, navigation issues, radio buttons accessibility issues and handling deep link users. The aforementioned concerns are not only related to the conformance with guidelines, but also to improving the interaction for disabled users.

Future work will address issues of user interface adaptation to the users needs, providing for example alternative layouts for each user category (e.g., blind or motor-impaired users). A possible approach would be to use different sets of templates for individual cases, which is easy and feasible, since presentation and behavior are independent from the data structure.

Furthermore, future work will include user-based evaluation, in order to test the accessibility of the system with the users it addresses to and obtain metrics not only regarding its accessibility but also regarding the efficiency and effectiveness of users with disabilities with the system. Therefore, even though the system is accessible, several interaction enhancements may be also implemented to achieve shorter task completion times and higher user satisfaction rates.

A barrier to the development process was the suggestion of W3C WCAG 1.0 to avoid the use of scripts, since in certain cases using them was necessary in order to achieve higher usability. For example, until user agents handle empty controls correctly, default, place-holding characters should be included in edit boxes and text areas. However, the absence of javascript forces the user to select the place-holding text and delete it before entering text. This could be automatically handled with the use of javascript, by deleting the place-holding text whenever the field obtains the mouse or keyboard focus. In general, the use of javascript could improve the usability of the application, providing an alternative context organization and shorter task completion times. Recognizing the need for adaptation to new technologies, W3C proceeded to the study and development of a newer version of its guidelines in order to include the new trends in web applications [11]. Therefore, future work will address the issue of evolving the conference management platform according to an Asynchronous JavaScript and XML (AJAX) approach.

References

1. World Wide Web Consortium Web Content Accessibility Guidelines 1.0.
<http://www.w3.org/TR/WCAG10>
2. May, M.: Integrating Accessibility into Content Management Systems. In: Proceedings of the XML Conference 2003, Pennsylvania, Philadelphia, USA (2003)

3. Burzagli, L., Billi, M., Gabbanini, F., Graziani, P., Palchetti, E.: The Use of Current Content Management Systems for Accessibility. In: Miesenberger, K., Klaus, J., Zagler, W., Burger, D. (eds.) ICCHP 2004. LNCS, vol. 3118, pp. 331–338. Springer, Heidelberg (2004)
4. Phpwebsite website <http://phpwebsite.appstate.edu>
5. Watchfire Bobby <http://www.watchfire.com/products/webxm/bobby.aspx>
6. Lynx text browser <http://lynx.browser.org/>
7. W3C Markup Validation Service <http://validator.w3.org/>
8. W3C CSS Validation Service <http://jigsaw.w3.org/css-validator/>
9. AccessColor - Online Tool for Colour Contrast <http://www.accesskeys.org/tools/color-contrast.html>
10. Colorblind Web Page Filter <http://colorfilter.wickline.org/>
11. Roadmap for Accessible Rich Internet Applications (WAI-ARIA Roadmap) <http://www.w3.org/TR/aria-roadmap/>
12. Nielsen, J., Loranger, H.: *Prioritizing Web Usability*, New Riders Press (2006) ISBN: 978-0321350312