A Software Solution for Accessible E-Government Portals

Dietmar Nedbal and Gerald Petz

Department of Electronic Business, School of Management,
Upper Austrian University of Applied Sciences,
Wehrgrabengasse 1-3, A-4400 Steyr, Austria
{dietmar.nedbal,gerald.petz}@fh-steyr.at

Abstract. The importance of e-Government web sites being accessible to people with disabilities is rising within the European Nations. Several European programs and state-laws are calling on governments and local government authorities to build accessible web sites. The paper deals with the development of a web content management system (CMS) that is already widely used by municipalities in Central Europe in accordance with the guidelines of the Web Accessibility Initiative (WAI). The project methodology is presented, followed by a discussion on the design of the database, the separation of layout and content and the implementation of the web site with Cascading Style Sheets (CSS).

Keywords: e-Government, accessible CMS, accessible software, WAI, triple-a.

1 Introduction

The European Nations are becoming more and more aware of the importance of accessibility in their e-Government services. The European Community has launched several programs like the "eEurope 2005 Action Plan" and its successor-program "i2010" [1]. These programs emphasize the importance of accessible Web services [2]. In Austria the "e-Government"-law (§ 1 Abs. 3) claims that web sites of governments and local government authorities have to be accessible in accordance with international standards (e.g. Web Content Accessibility Guidelines 1.0 (WCAG) of the Web Accessibility Initiative (WAI)) by 2008 [3].

The following paper focuses on the further development of the content management system (CMS) "RiS-Kommunal" in reference to web accessibility. This existing CMS is already widely used by local government authorities in Austria, Italy and Germany. It is the market-leading software in Austria – about 40% of Austria's municipalities use RiS-Kommunal. Thus the impact of upgrading the CMS to accessibility is significant, but several challenges arise as well:

(i) Web content management systems in general face the problem of limited WAI-ability; in many cases content authors have to have knowledge and awareness of accessibility.

(ii) The CMS mentioned above is not a "single website project"; approximately 1,000 systems are hosted on a central platform and because of this the complexity increases.

K. Miesenberger et al. (Eds.): ICCHP 2008, LNCS 5105, pp. 338–345, 2008.

(iii) Unfortunately nation-specific disability-related documents and laws have become effective and therefore each nation's specific guidelines have to be taken into consideration [4].

(iv) The CMS is a multi-lingual system; hence the compliance with WAI-guidelines becomes more complex.

(v) The ATAG guidelines (Authoring Tool Accessibility Guidelines) provide useful clues, but they do not ensure that WAI-compliant websites are produced.

Current Situation. Several studies concerning the adoption of accessibility guidelines can be found in literature ([5], [6], [7], [8]); the general result is that the majority of the web sites are not accessible. A report of the Federal Chancellery of the Republic of Austria reveals that the evaluated web sites have an averaged degree of performance of 94% WAI-compliance. However, this evaluation contained only 68 domains, and furthermore these domains are mainly provided by Federal ministries [9].

A different picture is presented in our own study of municipal web sites. The study was done by automated tests with the WAI-validator Cynthia (www.cynthiasays.com) in April 2006 and in January 2008 and comprised the 40 biggest cities and local government authorities in Austria. In 2006 only 18% of the evaluated websites were WAI-compliant (WAI-A or higher) and only 40% at the beginning of 2008 (see Fig. 1).

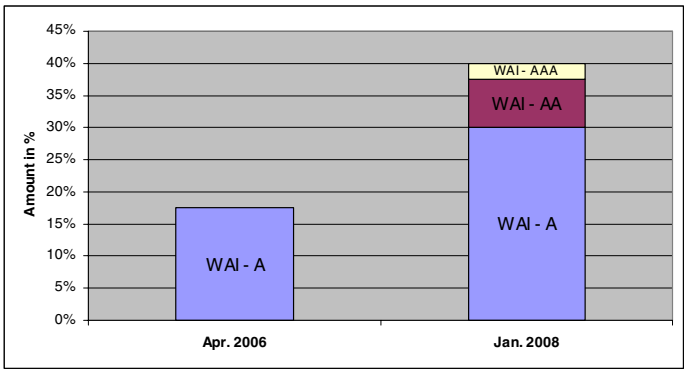

Fig. 1. Amount of WAI-compliant websites of Austria's 40 biggest cities

2 Methodology

The CMS follows the prototyping oriented lifecycle model [10]. Version 2.0 of the e-government solution was released in the year 2000. The specification at that time was primarily focused on multilingualism and structured data storage. Since the rollout of the software it has been maintained and continuously improved due to the customers' new requirements. The old presentation layer of the software was built on frames technology and nested tables were used for layout purposes. The software was programmed with classical active server pages (ASP code) and COM+-components. A link to a separate, text-only, alternative accessible page that provides equivalent information (WCAG guideline 11.4 [11]) was not an adequate solution for a state-of-the-art

accessible website. These facts lead to the decision of redeveloping the whole presentation front-end (including presentation layer, business logic layer and data access layer) and providing additional authoring tools for accessibility. The following phases were performed during the overall project:

(i) *As-is analysis:* The project started with the analysis of the running system, state-of-the-art literature research and analysis in the subject domain.

(ii) *Requirements definition:* The technical requirements from the as-is analysis, additional requirements from the user group, from accessibility guidelines across central Europe and from business environments were extracted and defined. A proven requirements catalogue was the result of this phase.

(iii) *Conceptualization:* In this phase, the requirements were consolidated, a roll-out concept, testing concept including beta-testing for support group and pilot projects and testing for early adopters were carried out.

(iv) *Training:* The trainings were performed in a pyramid scheme. All in-house programmers were trained by experts in the field of accessibility, the support group was trained by the senior programmer and the support group held trainings for the target group at an early stage to raise awareness of the accessibility topic among the municipalities.

(v) *Design:* The software was decomposed into modules and subdivided into design of the software presentation layer (layout and content), design of the business layer, design of the CMS and redesign of the data access layer and the database.

(vi) *Implementation:* Implementation was based on evolutionary prototyping. It started with an early user interface prototype showing a working Cascading Style Sheet (CSS) in combination with static XHTML throughout all major browsers. In parallel the data access layer was implemented and the business layer of the major modules were programmed and assembled. A prototype with all core modules was made available for beta testing to all support partners in a separate testing environment. Additional modules were added continuously and existing ones were tested and improved throughout the entire implementation phase.

(vii) *Installation:* The software was provided as an application service on a central server platform. As version 2.0 and the new accessible software solution was running on the same platform, the installation of version 3.0 did not have any effect on the running software instances of the municipalities.

(viii) *Update and migration*: Updates from version 2.0 and new installments were arranged by the support crew and activated for preparation. In the first wave a couple of "friendly users" were activated for version 3.0. Activation means that both versions of the software were running in parallel. At the beginning of the preparation a workshop concentrating on accessibility and on new features of the software release was held. The migration of the content and adjustment of the design took several days to several months to complete, depending on the amount of data and availability of the municipality's personnel resources.

(ix) *Evaluation and continuous improvement:* During operation, the results were evaluated and continuously improved to always ensure best-attuned software.

The phases did not necessarily follow a linear sequence; they overlapped each other and were cross-linked by feedback. Further important accompanying cross-the-phases aspects considered were: (i) documentation of the results; (ii) testing of all

components and modules; (iii) security issues; (iv) assuring high quality of software development; (v) availability of the software solution (both software releases had to operate on a 24x7 basis);

3 Cornerstones of Development

This chapter briefly discusses the crucial parts in the development of the accessible CMS. It starts with the design of the database and is followed by a deeper look inside the layout and the content preparation of the software solution. A quick overview of web accessibility evaluation tools concludes the chapter.

Designing the Database. Structured data is – of course – a prerequisite for a CMS. But many web content management systems provide only weak structured data, e.g. a simple article consists of only a few fields (headline, teaser, text) and in addition different contents are simply treated as an article.

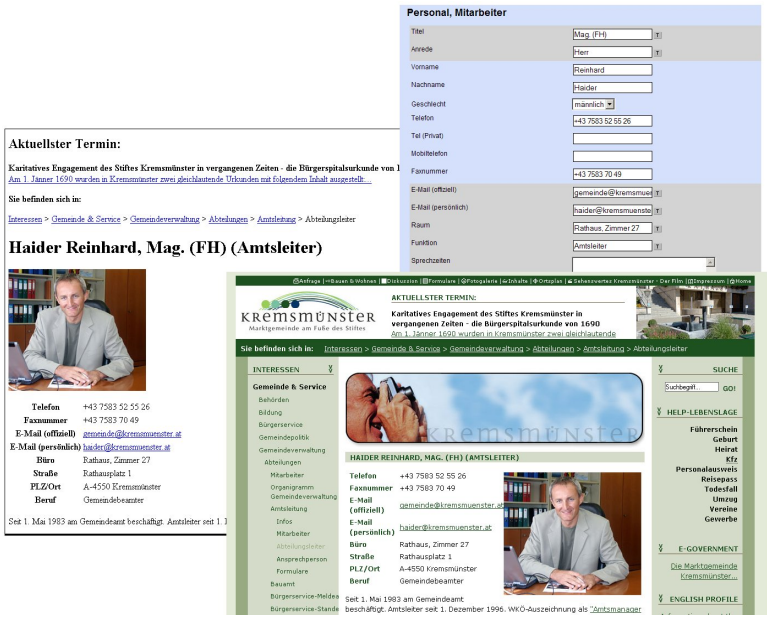

Fig. 2. Back-end of the CMS on the top right, front-end with disabled CSS on the left and front-end with enabled style sheet on the bottom right

Our database utilizes highly structured data and a predefined database structure for more than 50 modules especially designed for the needs of a municipality including administrative and political personnel, departments, regulations, responsibilities, forms, voting results and many more. Fig. 2 outlines the back-end of the CMS on the top right showing the attributes of a person. The database also distinguishes between language dependent information (like the title or the function that are highlighted in grey in the screenshot) and language independent information (e.g. surname or telephone).

When implementing the front-end of the software solution a strict separation of content and layout is one of the main paradigms. Therefore Cascading Style Sheets (CSS) were used to attach style to the content (CSS1 [12], CSS2 [13]). Fig. 2 also shows the front-end of the website with and without CSS and proves the fact that it can be easily linearized due to the structured database design and use of CSS for the layout. The only XHTML table on the website is used for the person's contact data (telephone, email, address).

The Layout. Both the CSS1 and the CSS2 standards are interpreted differently by existing browsers and this is discussed on various websites and in literature, simply searching the web for "CSS workarounds" also proves this. Browser inconsistencies and bugs like the box model muddle, the peekaboo bug, the guillotine bug or the three pixel text jog ([14], [15], [16]) just to name a few, took a lot of effort to circumvent. Avoiding tables for layout purpose and consequently using relative units ("em" and percentages) for sizing and positioning throughout the complete CSS made this even harder. Everyone on the web seems to be searching for the "one true layout" and is struggling with browser bugs and CSS hacking issues. This simple three set process for dealing with CSS problems should be applied in practice [17]:

1. If there is a known bug, it can sometimes be prevented.
2. If the bug cannot be prevented, a workaround should be used.
3. If the above process steps can not be applied, a hack is the only/last solution.

The main message is that a strategy has to be set up before using a hack. Hacks should be considered only as a last resort since they are browser-oriented solutions and there is no chance of predicting how the hack will behave on future versions of the specific browser. Workarounds on the other hand are CSS-oriented solutions to circumvent rendering problems.

Through intensive research and tests, we were able to create a basic layout template that is completely based on CSS standards without tables, only with the use of relative units, working with all current browsers and thus usable and adaptable for all municipalities. As shown in Fig. 3 the main layout was divided into eight parts where the users can place their boxes and content. The whole website can be limited to a maximum and a minimum width. For browsers that do not interpret the CSS2 elements "min-width" and "max-width" (i.e. Internet Explorer lower than version 7) an additional bit of JavaScript code was necessary. However, the web site can also be viewed when JavaScript is disabled. Horizontally the website is divided into three parts: header, content and footer. For the content a three column ("col-a, -b and –c") layout was chosen, whereas the third part can be left out to save space for the main content if desired. The column with the website content ("col-a") was additionally split into content-header, main-content and content-footer.

Having solved the main layout issues the next step was to provide common CSS layouts for the predefined content blocks like the hierarchical menu, the search box or any kind of textual and graphical content.

The Content. The content itself was structured by using XHTML tags like headings, subheadings, lists and tables, thus enabling the website to be displayed in a linear manner when disabling CSS or when browsing it text-only. Due to the highly structured database

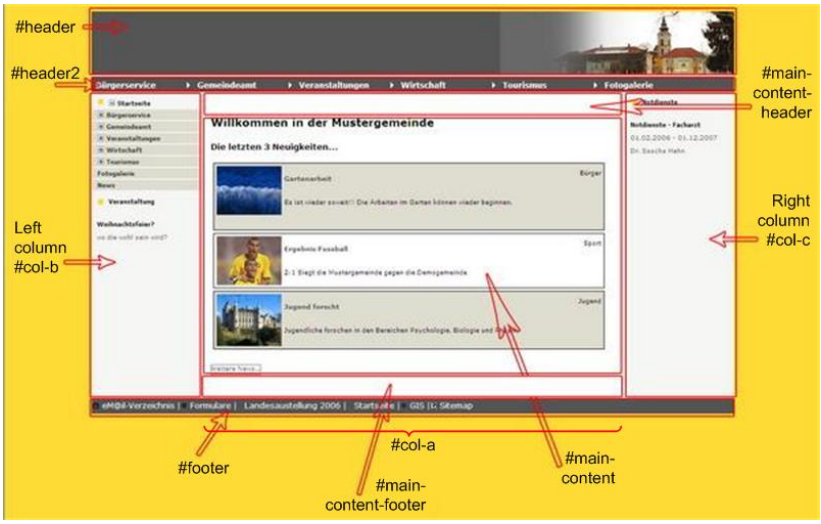

Fig. 3. Main CSS layout

the readout of the data (for example, the people of the municipal office) in accordance with the criteria of accessibility was actually possible and was achieved without having to change the existing data. Each content had to have at least one heading, subheadings were provided when necessary, enumeration of contents were displayed as (un)ordered lists and tables were only used to display tabular information like address or contact blocks and therefore always had a summary and a horizontal or vertical header.

Minor problems have arisen from the conformity with the XHTML standard for some special characters (like the encoding of the ampersand). In response to requests from the target group (for example, the attribute "target" for links was stipulated) finally the document type "XHTML 1.0 Transitional" was chosen as standard of conformity.

In order to provide easy and usable website navigation the software solution provides the following: (i) universal main navigation is structured as nested unsorted list; (ii) access keys are available for the most important links, including the possibility to skip the navigation and directly proceed to the content; (iii) the title of links can be set separately; (iv) an additional sitemap is available; (v) navigation via keyboard is possible; (vi) a breadcrumb provides information where the user is currently located within the website; (vii) a simple full text search feature is available to find the desired content directly; (viii) the whole website is operable without CSS and JavaScript. With the help of CSS the website administrators can decide whether the navigation is presented as a horizontal drop down, vertical drop down, explorer-like with expandable plus and minus signs or only by links.

In addition, the software supports the following features that enable and increase accessibility: (i) content-sensitive metatags and titles; (ii) alternative tags for images; (iii) accessible online forms that are conform to the Austrian e-government style guide [18]; (iv) primary language and changes to language can be identified; (v)

markup for abbreviations and quotations; (vi) website can be read without CSS and a high-contrast CSS is available; (vii) download documents are provided with a title; (viii) only data tables with an obligatory summary are used and headers are provided by the system.

Tools for Testing. In order to achieve WAI-compliance several tools turned out to be very helpful: Firefox Web Developer Toolbar, Lynx, Web Accessibility Toolbar, W3C-Validators (XHTML + CSS), Cynthia, WebXact, Taw, anybrowser and screen readers such as Jaws [19]. Automated testing-tools are not always appropriate and have certain limitations (e.g. changes in the natural language, consistent and usable structure and navigation, …). So we had during the implementation contact to experts as well as to people with disabilities to verify the correct and usable implementation of the WAI-guidelines.

4 Conclusions

Accessibility has to be faced from the outset as well as managed as a continuous process [20]. Although the system is capable of fulfilling all rules of the WAI guidelines, it is also possible to launch a portal that doesn't even conform to the lowest criteria of accessibility. Even if you foresee all technical possibilities, a willing and well trained website administrator is crucial for a sustainable accessible website. Therefore it is inevitable to raise awareness of the topic in general. The fact that accessibility raises the usability and the quality of a website should be emphasized, legal and governmental regulation should not be seen as the key driver for adoption of accessibility.

The CMS provides WAI-capabilities, but there is always room for improvement. Future developments should force web site administrators to adopt WAI-guidelines even more strictly at the expense of reduced usage of widely spread deprecated web-features, non-compliant, non-standardized tags and page-specific formatting. This will turn into a tightrope walk between requirements of the involved people. Beyond that, research has to deal with the following factors: (i) More and more web sites use Web 2.0-technologies such as AJAX to improve usability and user experience. (ii) User generated content is likely to offer poor accessibility because users are not aware of those issues. (iii) The new WAI guidelines (WCAG 2.0) are of wide scope and technology-neutral and interpreting these guidelines correctly will become more and more difficult. The question remains, how these new trends and technologies can be combined with accessibility standards to satisfy the requirements of all users.

References

1. Commission of the European Communities, i2010 – A European Information Society for growth and employment, `http://eurlex.europa.eu/smartapi/cgi/sga_doc?smartapicelexplusprodDocNumber&lg=en&type_doc=COMfinal&an_doc=2005&nu_doc=229`

2. Commission of the European Communities, Communication from the Commission to the Council, the European Parliament and the European Economic and Social Committee and the Comittee of the Regions - eAccessibility: [SEC(2005)1095], http://eurlex.europa.eu/smartapi/cgi/sga_doc?smartapicelexplusprodDocNumbe&type_doc=COMfinal&an_doc=2005&nu_doc=425&lg=en
3. Bundesgesetz über Regelungen zur Erleichterung des elektronischen Verkehrs mit öffentlichen Stellen (E-Government-Gesetz - E-GovG) (January 23, 2008), http://ris1.bka.gv.at/authentic/findbgbl.aspx?name=entwurf&format=html&docid=COO_2026_100_2_30412
4. World Laws: Europe (January 04, 2008), http://www.webaim.org/articles/laws/world/europe.php
5. Condit Fagan, J., Fagan, B.: An accessibility study of state legislative Web sites. Government Information Quarterly 21(1), 65–85
6. Shi, Y.: The accessibility of Chinese local government Web sites: An exploratory study. Government Information Quarterly 24(2), 377–403
7. Hackett, S., Parmanto, B., Zeng, X.: A retrospective look at website accessibility over time. Behaviour & Information Technology 24(6), 407–417
8. Website Survey October 2007 - Local Government - UK, (November 14, 2007), http://survey-beta.sitemorse.com/survey/report.html?rt=278
9. Eibl, G., Wagner-Leimbach, H.: Erhebung Barrierefreiheit, Endbericht (2007)
10. Pomberger, G., Blaschek, G.: Software-Engineering: Prototyping und objektorientierte Software-Entwicklung, 2nd edn. Hanser, München (1996)
11. Chisholm, W., Vanderheiden, G., Jacobs, I.: Web Content Accessibility Guidelines 1.0: W3C Recommendation 5-May-1999 (December 12, 2007), http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/
12. Bos, B., Wium Lie, H.: Cascading Style Sheets, level 1: W3C Recommendation (December 17, 1996), revised (January 11, 1999) (December 12, 2007), http://www.w3.org/TR/1999/REC-CSS1-19990111
13. Bos, B., Wium Lie, H., Lilley, C., Jacobs, I.: Cascading Style Sheets, level 2: CSS2 Specification W3C Recommendation (May 12, 1998) (December 12, 2007), http://www.w3.org/TR/1998/REC-CSS2-19980512
14. Bergevin, H., Gallant, J.: Position Is Everything: Modern browser bugs explained in detail! (December 17, 2007), http://www.positioniseverything.net/
15. Sawyer McFarland, D., Lunsford, K.: CSS Troubleshooting. Macworld 22(2), 84–85
16. Schmitt, C., Lang, J.W.: CSS-Kochbuch: [schnelle Lösungen für typische CSS-Probleme], 2nd edn. O'Reilly, Beijing (2007)
17. Holzschlag, M.: Integrated Web Design: Strategies for Long-Term CSS Hack Management (December 17, 2007), http://www.peachpit.com/articles/article.aspx?p=170511
18. E-Government Styleguide für E-Formulare 2.0.0, 6.9.2006 (January 09, 2008), http://reference.e-government.gv.at/Styleguide___stg_2_0_0_-_Versi.940.0.html
19. Luque Centeno, V., Delgado Kloos, C., Arias Fisteus, J., Alvarez Alvarez, L.: Web Accessibility Evaluation Tools: A Survey and Some Improvements. In: Proceedings of the International Workshop on Automated Specification and Verification of Web Sites (WWV 2005). Electronic Notes in Theoretical Computer Science, vol. 157(2), pp. 87–100 (2005)
20. Jaeger, P.T.: Assessing Section 508 compliance on federal e-government Web sites: A multi-method, user-centered evaluation of accessibility for persons with disabilities. Government Information Quarterly 23(2), 169–190