

Adapting the Web Development Environment for Accessible and Usable Application Construction

Elena García Barriocanal¹

Miguel Ángel Sicilia Urbán
Computer Science Department. Carlos III University.
Polytechnic School. Av. Universidad, 30. Leganés (28911). Madrid. Spain
msicilia@inf.uc3m.es

José Ramón Hilerá¹

¹Computer Science Department. Alcalá de Henares University.
Polytechnic School. Campus Universitario s/n. Alcalá de Henares (28871). Madrid. Spain
{[elena.garciab](mailto:elena.garciab@uah.es), [jose.hilera](mailto:jose.hilera@uah.es)}@uah.es

Abstract: Web application accessibility requires the enforcement of a set of rules on the content written or generated with the help of Web Development Environments. W3C Consortium's *Web Accessibility Initiative* is the most significant normative effort in this direction, and provides an evolving corpus of accessibility recommendations for HTML and, in the future, for other Web languages. In this paper, we describe a practical and extensible approach to integrate those recommendations in commercial Web Development Environments by using a rule database.

Introduction

Web application accessibility can be defined as the application capability of being used by the widest range of users (disabled, elder, etc.). The World Wide Web Consortium (W3C) promotes the Web Accessibility Initiative (WAI) as a forum for discussion and normalization of Web accessibility practices. The WAI makes available a number of evolving recommendations that apply to different elements of the Web architecture. Currently, the most important ones for our purposes are the web content guidelines (W3C 1999) and the authoring tool guidelines (W3C 2000). Most Intranet/Internet browser-based applications being developed for enterprise internal use or Web e-Commerce are constructed with the help of a *Web Development Environment* (WDE), e.g. Microsoft Visual InterDev or Macromedia Dreamweaver UltraDev. We will refer to the product of a WDE as *Web Application* (WA), different in purpose and use from a conventional web site made up exclusively of static content. We claim that the WDE is one of the most important elements to be adapted for accessibility, since its products are the applications that will be present in most of the business that require the use of computer applications. The work presented here is motivated by previous analysis of the importance of this point (García B. & Sicilia U. 2000a), (García B. et al. 2000b). In the rest of this paper, we briefly describe generic techniques for adapting a WDE to build accessible applications.

WDE Adaptation for Accessibility

By analyzing common functions in commercial WDEs, we have identified three key adaptation areas: (a) content, generated or written directly by the developer, (b) prepackaged content generated by the tool, and (c) help systems and wizards. The core component of our WDE adaptation layer (a generic name for the components added or modified) is called the *content rule-checking engine* (CRE), designed to cover the first area previously described. This module operates on WA content directly (whenever developers require it, by pressing the corresponding button) or handling some events from WDE editors. The adaptation layer can respond to those events by parsing the last line typed and taking some corrective action (showing a dialog box, for example) if some inaccessible element has been inserted. The CRE takes content fragments as input, examines them, and can return a corrected version or trigger some kind of program to correct it. The *checker* module, one of the elements inside the CRE, views

HTML/XML documents in a tree-like form, as exposed by Document Object Model (DOM)-based parsers (W3C 1998). A content rule database defines accessibility constraints on those trees. According to the rules, the *checker* modifies the HTML tree and saves it back directly or after asking for some kind of user information via dialogs (activated through some sort of *dialog executor*). Those dialogs are stored in a separate component repository in some binary or script form, e.g. ActiveX components in the Microsoft Windows platform.

The design of the rule database used by the *checker* was the result of the process of classifying WAI content checkpoints, depending on the structural part of HTML involved and the kind of corrective action needed. A rule applies on a subject, either a specific HTML item or a relationship between two of them. A detailed analysis of the Web Content Accessibility Guidelines (WCAG) checkpoints suggests the following rule subjects: single elements, attributes in the context of an element, pairs of elements and collections of attributes in the context of an element.

The kinds of corrective actions that can be applied when a rule fires are four: (a) *ensure*, to enforce the user to include an element, an attribute or some content into the element, according to rule structure; (b) *avoid*, to forbid an attribute, element or structural relationship; (c) *suggest*, to prompt the user about some recommended but no essential accessibility aspect (some WCAG priority level 2-3 checkpoints can be modeled this way) or (d) *run external algorithm*, to execute an external program to check some accessibility aspect.

In addition, some rules apply to an element only if some condition holds. This introduces the concept of *applicability conditions* (predicates that discriminate if a rule fires or not in a specific context).

Finally, when the application of a rule requires the execution of some kind of program or dialog, we must store its reference to invoke the dialog executor.

The ideas presented here have been tested in Microsoft Visual InterDev 6, as an example of adapted WDE. The adaptation layer has been modified in three ways: (a) by installing an *Add-In* developed in Visual Basic 6 in the InterDev's toolbar, (b) by adding accessible *themes* and *layouts* as prepackaged content for new WA (this adaptation is as simple as adding a new entry in some configuration files and deploying some accessible HTML templates in InterDev directories), and (c) by including a browser-based context-sensitive help for *Add-In* windows, using HTML-formatted WAI documentation.

The *Add-In* checks some rules inferred from WAI checkpoints and makes use of a set of COM components that implement user interface windows to give some advice about accessibility or gather some input from the user.

Conclusions

The quality of the adaptation layer can be measured by the increment in conformance of the WDE regarding WAI recommendations (W3C 2000) and (W3C 1999) after adaptation. In our InterDev prototype, there are three significantly enhanced areas: (a) *Add-In* dialogs incorporate context-sensitive help, and a general help section about WAI web contents guidelines is accessible from *Add-In* main window, (b) our prototype is completely integrated in the overall "look and feel" of the WDE, since the *Add-In* installs itself in InterDev's tool bar with the same appearance of any other utility, and (c), previously-checked prepackaged navigation *layouts* and *themes* allow the use of navigation mechanisms in a consistent manner. Once analyzed the underlying rules in accessibility recommendations, building adaptation layers is a relatively easy and inexpensive development. HTML parsing programs are available for free and a WA is ultimately a set of HTML pages with mixed programming code. Usability and accessibility techniques for the Web have evolved in continuous feedback and most of them are similar or complementary, and therefore the techniques presented here can also be used to enhance WA's usability.

References

1. García B., E., and Sicilia, M.A. (2000a). Analysis of Internet accessibility problems from the point of view of employment. *Congreso Iberoamericano de Informática Educativa Especial2000*. Córdoba University Publishing Service, Córdoba (Spain).
2. García B., E., Sicilia, M.A., Hilera, J.R. (2000b). The Intranet as an enterprise application infrastructure. A new barrier for disabled people?. *III Jornadas de Informática y Sociedad*. Comillas Pontifical University Publishing Service, Madrid (Spain).
3. W3C. (2000). Authoring Tool Accessibility Guidelines 1.0 (ATAG). *W3C Recommendation*.
4. W3C. (1998). Document Object Model (DOM) Level 1 Specification Version 1.0. *W3C Recommendation*.
5. W3C. (1999). Web Content Accessibility Guidelines 1.0 (WCAG). *W3C Recommendation*.