

# ADIS: Accessibility and Device Independence System

Regina Bernhaupt  
ICT&S Center  
University of Salzburg  
5020 Salzburg, Austria

Leonhard Brunauer  
ICT&S Center  
University of Salzburg  
5020 Salzburg, Austria

Manfred Tscheligi  
ICT&S Center  
University of Salzburg  
5020 Salzburg, Austria

Regina.Bernhaupt@sbg.ac.at

lbrunau@cosy.sbg.ac.at

Manfred.Tscheligi@sbg.ac.at

## ABSTRACT

ADIS (Accessibility and Device Independence System) is a system capable of generating accessible and device independent hypertext structures out of a single data repository. The system helps authors and maintainers of content in hypertext structure to handle accessibility and device independence without much effort. To achieve accessibility of websites ADIS computes an accessible version for every webpage. Device independence is achieved by means of content selection according to the client's capabilities. ADIS additionally delivers an accessible version which can be used on all devices.

ADIS is based on several free available tools and can be used to flexibly generate accessible and/or device independent content. A first version of ADIS is available in June 2005, the system will be finalized in December 2005 – thus ADIS is still work in progress.

## Categories and Subject Descriptors

H.5.4 [Hypertext/Hypermedia]

## General Terms

Design.

## Keywords

Accessibility, device independence, Lenya, Cocoon, mobile devices, interactive TV.

## 1. INTRODUCTION

ADIS (Accessibility and Device Independence System) is a tool to automatically and dynamically generate accessible and device independent versions of websites. ADIS was designed with three objectives in mind: first, making websites accessible to people with special kinds of visual impairments, second, making websites available for mobile devices and services on interactive TV and third, to generate an accessible version for all supported devices (accessibility and device independence combined). The functionality of ADIS is limited, though. ADIS' applicability lies in web projects focusing on presentation of content – i.e. projects where user interaction is only required to a certain extend. In particular, ADIS should not be considered as a platform for creating web applications.

### 1.1 Accessibility with ADIS

Making websites accessible to people with special kinds of visual impairments, using for instance a screen reader to gain access, can be addressed in various ways. Following checklists [7], sticking

to guidelines [4] and using technical tools [2], [6] can help to improve the accessibility of websites. ADIS' goal is to make it as easy as possible for authors to produce accessible content according to these recommendations. An emphasis was put on compliance to the Authoring Tool Accessibility Guideline ATAG [11], though some of the guidelines are not met in the current implementation. In particular, the system is intended to support authors in writing proper content, which can in turn be transformed to be accessible.

But making websites – or more generally XML based content – accessible is just the first step. The other major focus of ADIS is to present content device independent. Although this is often realized on the client side – see [3] for an example targeted at mobile devices – ADIS uses a different approach. In ADIS the complete presentation generation is done on the server side. Device independence is achieved by means of content selection according to the client's capabilities.

Additionally, we aim to go one step further and use ADIS not only to make content available on various devices but to be available in an accessible version. For example, the accessible version of a website can be viewed using a PDA. In some cases, of course, using the accessible version can be out of scope. Another focus while designing ADIS' device independence features was put on interactive TV.

### 1.2 Device Independence

We achieve device independence on the server side by selecting the content according to the client's capabilities. The mechanisms to get these properties are based on the W3C's Composite Capability/Preference Profile CC/PP [9] and the Open-Mobile-Alliance group's User Agent Profile UAProf recommendation. There are several types of properties in both specifications, where ADIS utilizes those related to software and hardware. Hence, user preferences are not taken into account in the current version.

To enable selection of text based content, ADIS asks the authors to provide summaries for longer text sections – e.g. articles. These summaries are selected for devices with small screens or very low bandwidth. This means that the content source for these devices differs from the one delivered to other devices. But since the two versions are stored within the same content source file, authors always see both of them in the editing area, which makes it somehow possible to keep them synchronized.

With the exception of distinct sources for summary and full text sections, all the different output versions are generated out of the same content source, which makes the source format an important issue. One of the requirements is to have a language with

structural tags only. This is mandatory for the content selection process. Since the formatting is done with respect to the generated output version, it is the transformation's job too. Thus, formatting is left out of the content. XHTML2 meets our demands with the additional advantage of being easily transformable to XHTML1.

Currently, modifying content requires the author to know XHTML2, because we use a plain-text editor. To be broadly accepted, a user-friendly editor will be added in June 2005. Furthermore, it should support authors in writing content that conforms to the Web Content Accessibility Guidelines WCAG [4] as stated in ATAG Guideline 3 [11]. Some extensions to this guideline are made to ensure creation of device independent content. For instance to create the above mentioned summary sections.

### 1.3 Advantages of ADIS

Several projects have been focusing on the development of tools to enhance accessibility or to enable device independence. ENABLED [10] is developing a toolkit to support the transformation of non-accessible web content into accessible forms and multimodal representations. Moreover, the project addresses mobile aspects. On the other side the project IRIS [5] is building a tool to give design recommendations for producing accessible websites. With respect to these projects ADIS can

- produce accessible versions of websites dynamically. For example we do link a so called visual version to its accessible counterpart (see figure 1). The accessible version enhances the website with a glossary, with a special navigation supporting screen readers and it gives the possibility to dynamically change font size (like suggested in [6]).
- produce not only device-independent websites, but additionally deliver an accessible version which is linked on every website. A disabled user in front of a TV screen thus can easily use the accessible version for interactive TV.

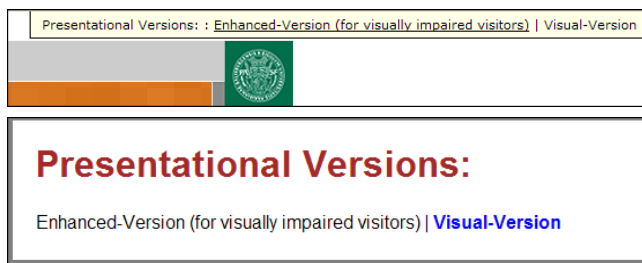


Figure 1: Switching between the standard visual and enhanced accessible version

## 2. IMPLEMENTATION

ADIS is based on the Apache Lenya CMS (<http://lenya.apache.org>) and the Cocoon publishing framework (<http://cocoon.apache.org>). The goal of ADIS is to support accessible and device independent versions of XML based content. ADIS is currently under development and is used in a first version to create a fully accessible version of a website.

To make websites accessible, ADIS must perform a number of transformation steps before content is delivered to the client. XSLT is used to implement these transformations.

### 2.1 Components and Add-Ons

The Apache Lenya CMS was the perfect solution to build on. Lenya is easily extensible by – what its authors call – ‘publication types’. Maybe the greatest advantage of Apache Lenya is that content is not required to be in a single format, like XHTML or HTML. Any valid XML based language can be added to the content resource types, which makes our work a lot easier. Lenya is built upon the Cocoon framework, which provides an XSL transformation pipeline. This is the place where we added our stylesheets. ADIS is an adaptation of Lenya’s default publication type with customization of the content resource format, new transformation stylesheets, and modifications in the outline of the Cocoon pipeline.

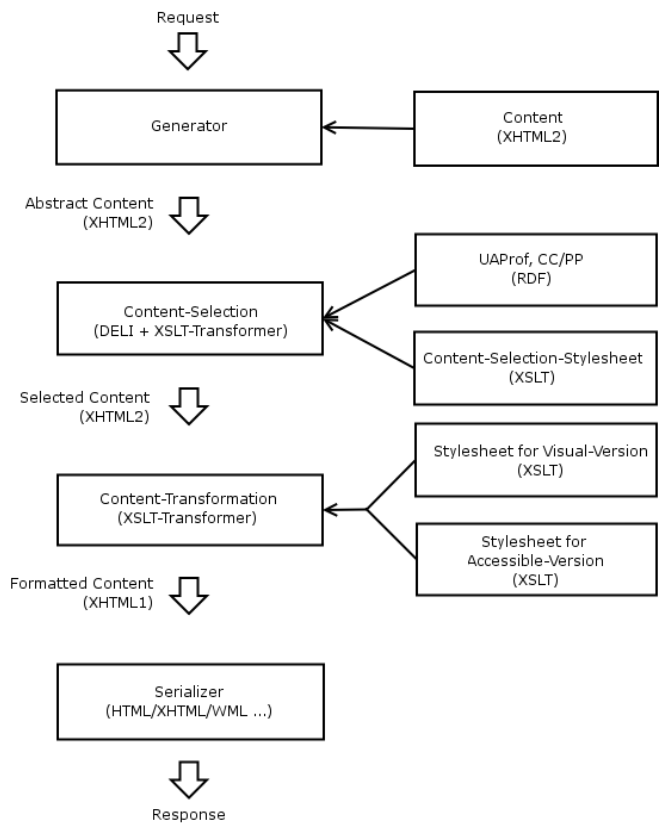


Figure 2: Transformations to generate accessible and device independent content dynamically

The transformation from content to output is laid out as a pipeline (see figure 2). Upon request, a ‘generator’ reads the appropriate content file and sends SAX events down the pipe. This step can be thought of feeding the pipeline with content. To achieve device independence, the relevant parts of the content source must be selected, which is done in the first transformation step. In the next step another transformation is performed which either creates a visually enhanced or an accessible version. The translation from XHTML2 to XHTML1 is also done in this part of the pipeline.

After this step the transformation process is finished. The transformation to XHTML1 is performed because the pipeline's end – which is called a 'serializer' – needs XHTML1 input. The serializer takes its input – which are still SAX events – and translates it to the desired output format. Its output is in turn delivered to the client.

The stylesheets for the accessibility processing step are based on [8]. In fact there are two distinct transformations, where the client chooses which one is carried out. One of them issues a version with enhanced accessibility, while the other one creates a standard visual version. ADIS analyzes the requested filename and uses the appropriate stylesheet for the filename suffix – e.g. the visual-version stylesheet for `index.vis.html`. A banner at the top of each page enables users to switch between visual and accessible version.

The differences between the two presentational versions are on the one hand structural ones. This includes different navigation mechanisms, glossary, and support of special key-strokes for the accessible version. On the other hand the versions are formatted differently. The accessible version features larger standard fonts, greater contrast, and clear boundaries of structural entities.

As already mentioned, ADIS must select the content that is relevant to the client. This step obviously depends on the client's capabilities like bandwidth, screen size, etc. But the server has to get this information somehow. Fortunately, such a tool has already been integrated with Cocoon. The DELI [1] add-on delivers these capabilities to the transformation process. When a client device is recognized, the properties defined in the appropriate CC/PP or UAProf are available to the content selection stylesheet. Usually, clients provide a link to their profile in the HTTP request. In case they don't, DELI maintains a repository where user-agent-profiles (UAProf files) are stored. This makes it easy to add new devices to ADIS by putting a proper UAProf file to the repository. Then DELI must be told which user agent string (i.e. the string sent along with the HTTP request) should be mapped to which profile. This feature helps ADIS recognize interactive TV or other non CC/PP-aware devices without modifying the client's software.

### 3. SUMMARY

ADIS is a system enabling users to get accessible versions of websites. Authors are supported in creating accessible websites by reminding them to stick to guidelines. The system is not limited to generate accessible websites, but additionally enables the preparation of content to be displayed on different devices like mobiles or TV screens. However, the most outstanding feature of ADIS is the support of accessible versions of websites on different devices. That is, for example, accessible versions of websites can be "viewed" on TV screens.

### 4. FUTURE WORK

In June 2005 the first step of ADIS was accomplished providing the functionality to build accessible websites. A focus is set on making ADIS fully satisfying the ATAG 1.0 [11] recommendations, which implies an accessible interface for authors. The next step will be the integration of device independence, which will be finished in September 2005. ADIS will be further enhanced by a special user interface for the CMS and will be tested on several occasions especially within a project on interactive TV. In the next development cycle ADIS will get an accessible and user-friendly interface supporting the user in editing the CSS stylesheets and modifying the transformation stylesheets. The final version of ADIS shall be available in December 2005.

### 5. REFERENCES

- [1] Butler, M.H. DELI: *A delivery context library for CC/PP and UAProf*. External Technical Report HPL-2001-260, 2001.
- [2] CAST Bobby. Available at <http://www.cast.org/bobby/>
- [3] Chen, Y., et al. *Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices*, WWW 2003, Budapest, 2003.
- [4] Chrisholm, W., Vanderheiden, G., and Jacobs, I. *Web Content Accessibility Guidelines 1.0*. 1999. Available at <http://www.w3.org/TR/WCAG10/>
- [5] Darzentas, J. et al. *IRIS: Implementing an Open Environment supporting Inclusive Design of Internet Applications*, Interact03, Zürich, 2003.
- [6] Hanson, V. *The User Experience: Designs and Adaptions*, WWW 2004, New York, 2004.
- [7] IBM Web Accessibility Checklist. Available at <http://www-3.ibm.com/able/guidelines/web/accessweb.html>
- [8] Katstaller, G. *Accessibility als leitender Faktor bei der Website Entwicklung*. Internal Report, University of Salzburg, Austria, 2005.
- [9] Klyne, G., Reynolds, F., Woodrow, C., Ohto, H., Hjelm, J., Butler, M.H., and Tran, L. *Composite Capability/Preference Profiles (CC/PP)*. 2004. Available at <http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115>
- [10] Project ENABLED. Available at <http://www.enabledweb.org/>
- [11] Treviranus, J., McCathieNevile, C., Jacobs, I., and Richards, J. *Authoring Tool Accessibility Guidelines 1.0*. 2000. Available at <http://www.w3.org/TR/2000/REC-ATAG10-20000203>