A Flexible Software Architecture Concept for the Creation of Accessible PDF Documents

Alireza Darvishy, Hans-Peter Hutter, Alexander Horvath, and Martin Dorigo

ZHAW Zurich University of Applied Sciences
InIT Institute of applied Information Technology
P.O. Box 805
CH-8401 Winterthur, Switzerland
alireza.darvishy@zhaw.ch

Abstract. This paper presents a flexible software architecture concept that allows the automatic generation of fully accessible PDF documents originating from various authoring tools such as Adobe InDesign [1] or Microsoft Word [2]. The architecture can be extended to include any authoring tools capable of creating PDF documents. For each authoring tool, a software accessibility plug-in must be implemented which analyzes the logical structure of the document and creates an XML representation of it. This XML file is used in combination with an untagged non-accessible PDF to create an accessible PDF version of the document. The implemented accessibility plug-in prototype allows authors of documents to check for accessibility issues while creating their documents and add the additional semantic information needed to generate a fully accessible PDF document.

Keywords: Document accessibility, automatic generation of accessible PDF, screen reader [3], visual impairment, accessibility, tagged PDF, software architecture.

1 Introduction

There are thousands of PDF documents available on the internet, but most of them are not accessible for visually impaired people using screen readers [3]. In many cases authors use authoring tools such as Adobe Indesign [1], Microsoft Word [2] and others to create these PDF documents. But all too often the resulting PDFs are not tagged correctly and have to be manually post-processed in order to become accessible PDFs. This is inefficient, very time-consuming, and tedious. In addition, a separate solution is needed for each authoring tool because there is no software solution that can be used with different authoring tools.

The following list outlines typical issues that arise when creating a PDF document with an authoring tool such as Microsoft Word [2]:

- Table layouts: If tables are used for layout purposes only and not to present tabular data, the author of the Word document is not alerted to the fact that the information should be formatted with style sheets rather than in table layouts.
- Table headers: Word only allows for the creation of column headers – the Word author has no option to define row headers.

K. Miesenberger et al. (Eds.): ICCHP 2010, Part I, LNCS 6179, pp. 47–52, 2010.

- Form fields: The specified entry fields in Word are not carried over into the PDF.
- The Word author is not alerted if no alternative text is entered for a picture.
- The Word author is not alerted if the document is not structured with headings, or if standard style sheets are not used in Word for structuring purposes.
- The Word author cannot indicate the correct reading order.
- The author is not alerted if there is insufficient contrast between text and background.

Although newer versions of Microsoft Word [5] provide facilities to overcome some of the above-mentioned issues, this solution can only be used with MS Word [2] but not with other authoring tools. This paper introduces a new flexible software architecture approach that enables accessible PDF documents to be created from a variety of authoring tools.

2 Our Approach

The diagram in Fig. 1 shows the structure of the suggested architecture for MS Word [2].

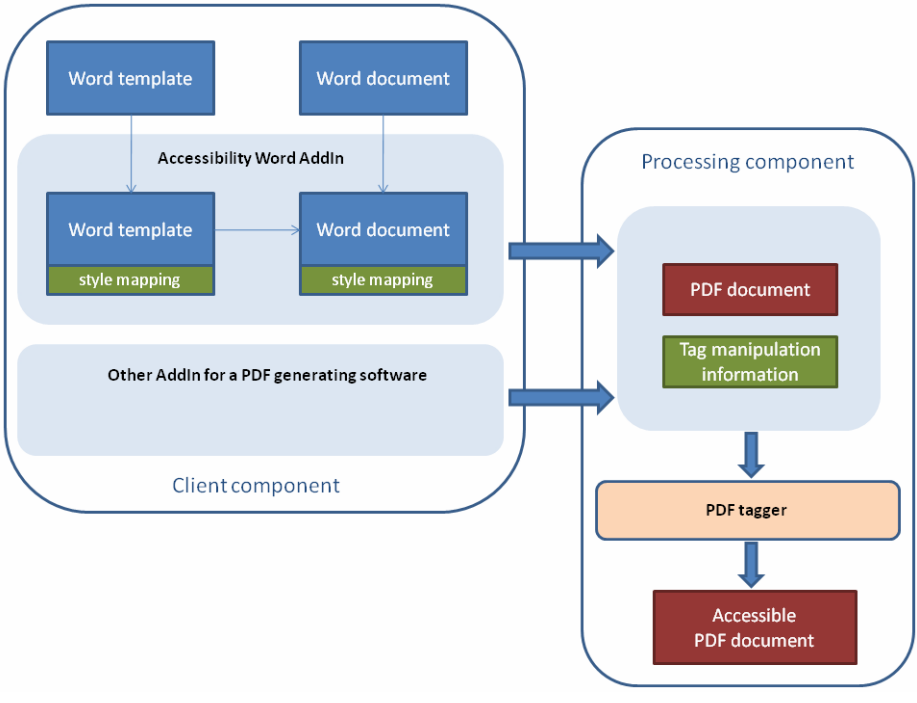

Fig. 1. Proposed software architecture for the generation of accessible PDFs

The architecture illustrated above consists of two components which can be implemented together on the authoring tool side or separately as client and server components:

2.1 Client Component

Fig. 2. Word Ribbon of Accessibility Add-in

This client component (authoring tool plug-in, see Fig. 2) executes the following steps:

- Accessibility Repair Tool (Fig. 3): Checks for accessibility problems associated with defined contents like headers, pictures, tables etc. and enables the author to correct them simply and easily.

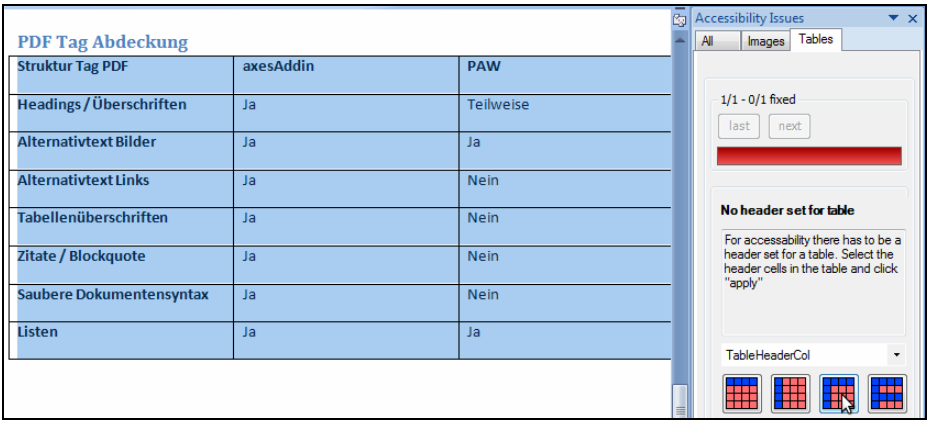

Fig. 3. Repair tool for table header issues

- Mapping (Fig. 4): Mapping of style sheets defined in an authoring tool (e.g. MS Word) to PDF tags. These mapping settings are then stored in the source document (in this case a Word document). The illustration shows, for example, that the "TableHeaderRow" style maps to the "Table header" tag in the PDF.

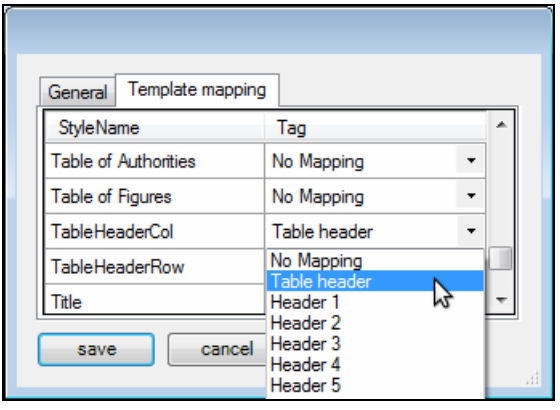

Fig. 4. Word style mapping to PDF tags

Based on the steps described above, an untagged PDF document and an XML document are generated. The XML document contains the structural elements (e.g. tables, form fields etc.) and their position in the Word document. This information is then used to tag the PDF document in the <Processing Component>.

2.2 Processing Component

This component uses the structural elements from the XML document and their positions to set the right tags in the PDF document (see Fig. 5).

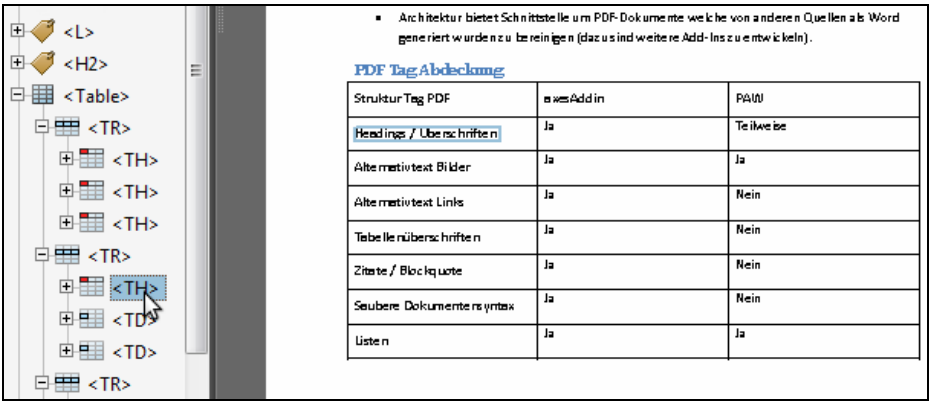

Fig. 5. PDF table structure tags

3 Use Cases

The following scenarios are planned for the use of the implemented software:

3.1 Use Case 1: Publishers

In large companies web publishers, among others, are responsible for generating templates. By using the client component described above web publishers can generate templates that are fully accessible. These can then be locked so that end users cannot subsequently modify them. This means that corporate identity and accessibility standards can be implemented without any need for end users to concern themselves specifically with the accessibility aspects.

3.2 Use Case 2: End Users

For small and medium-sized enterprises the implemented software can help end users to identify and resolve accessibility problems.

4 Existing Solutions

Tools that enable the production of tagged PDFs can be placed in the following categories:

1. Accessibility user support: Tools which in principle enable the production of tagged PDFs but which offer authors little or no help in identifying and dealing with accessibility issues when producing the source documents. These include MS Word [2], Open Office Writer [6] and Adobe Acrobat Professional [7].
2. Integrated pdf tagging component: Tools that have an integrated component for producing tagged PDFs and therefore do not require the installation of any add-ins.
3. Direct production of tagged PDFs: These tools come from Adobe inc. and require a license.

For almost all categories it is essential that the author be familiar with the accessibility requirements and knows how to put these into practice in the tool in question.

One exception is:

- NetCentric PDF Accessibility Wizard for MS Office [4] Accessibility problems are highlighted with the support of a software assistant. The author of the Word document has the option of correcting the issues highlighted. The PDF tags are saved locally through the allocation of styles to the structured information and are therefore not incorporated into the document itself. The software is not language-sensitive, so for example the headings are recognized in a German document even if they are defined in English (e.g. Heading 1). This wizard is only available for MS Office.

5 Conclusion

This paper presents a new concept for a software architecture that enables the automatic generation of fully accessible PDF documents. The suggested software architecture enables a reusable infrastructure that can be used with any authoring tool to create accessible PDF documents. The implemented software prototype for MS Word [2]

using an accessible PDF Word add-in is designed to resolve the issues mentioned in the introduction section of this paper. On the one hand the existing Word document is reviewed and assistance is offered to eliminate problems during the creation process. On the other hand, weak points that cannot be eliminated within Word [2] itself are analyzed and then eliminated using the external PDF tagger following export to PDF. The add-in can also be used to save documents as templates, whereby the structural information assigned to the user-defined styles is saved directly in the Word [2] file itself. Locked templates generated in this way can guarantee to a great extent the accessibility of documents created using these templates.

References

1. Adobe InDesign, http://www.adobe.com/products/indesign/
2. Microsoft Office Word, http://office.microsoft.com/word/
3. Freedom Scientific JAWS for Windows Screen Reading Software, http://www.freedomscientific.com/products/fs/jaws-product-page.asp
4. NetCentric PDF Accessibility Wizard for MS Office, http://www.net-centric.com/products/PAW.aspx
5. Microsoft Office (2010), http://us1.office2010beta.microsoft.com
6. Open Office Writer, http://www.openoffice.org
7. Adobe Acrobat Professional, http://www.adobe.com/products/acrobat