A Semi-automatic Support to Adapt E-Documents in an Accessible and Usable Format for Vision Impaired Users

Elia Contini, Barbara Leporini, and Fabio Paternò

ISTI-CNR, Pisa, Italy
{elia.contini,barbara.leporini,fabio.paterno}@isti.cnr.it

Abstract. Electronic material (e-documents, e-books, on line resources, etc.) represents an essential tool for continuous learning for print-impaired people, provided it is well-structured. To obtain accessible and usable e-content, specific requirements should be applied from the early beginning or used when adapting existing electronic formats. In this paper we present a method, and a first associated prototype, for making e-documents in a format, which is accessible and usable for vision impaired users. The resulting environment is composed of various transformations, with different degree of automation, and applies a number of guidelines that have been defined for this purpose.

Keywords: accessible publishing, e-books, accessibility, tool.

1 Introduction

In several countries, specific accessibility laws have been approved, which require publishers to provide electronic accessible versions of school textbooks. Nevertheless, disabled students in mainstream classrooms do not always have access to the same learning tools as their classmates. The time-consuming process of turning books or other materials into Braille, audiotape, or large-print editions means that visually impaired students often start the school year without their textbooks.

We use the term e-document for a content (text, figures etc.) that is stored in electronic form. Thus, an e-document can be a simple text, a more complex document, with title, paragraphs and subparagraphs, or an electronic book organised into chapters and sections. The format of many e-documents available to visually impaired people can be fairly simple and easy to modify to provide accessibility, as is often the case for narrative books. Unfortunately, it is not the same for the scientific e-documents, which may contain complicated explanations, graphs, tables and formulae. In this case it is more difficult to create accessible and usable texts. Overall, our work aims at addressing the accessibility and usability issues in electronic publishing, by providing recommendations for creating accessible e-documents and supporting the conversion of inaccessible e-documents.

In this paper, we first introduce the issues related to e-document accessibility and explain the basic motivations driving the research on this topic. A short section reports on the state-of-art in this field. Then, after having introduced the accessibility and usability issues that should be considered in designing e-documents for readers

K. Miesenberger et al. (Eds.): ICCHP 2008, LNCS 5105, pp. 242–249, 2008.

with vision impairments, we discuss main issues concerning e-book manipulation. To address this topic we propose a semi-automatic tool, currently in the prototype stage, to support the conversion of e-documents to an accessible and usable format.

2 Related Work

In order to overcome limitations for those people who cannot meet their information needs from standard printed material, several attempts have been made to develop specialised electronic formats. One of these is a by-product of the Daisy consortium (Digital Accessible Information SYstem) [4], which is working on producing a standard for talking books characterized of navigational information to better move through the document. With DAISY format hybrid books can be made available in the DAISY format, ranging from audio-only, to full text and audio, to text-only.

Other studies on this topic include the work by Sun et al. [8], who designed e-Book browsers that aid users in perceiving and understanding the important conceptual structures of a book, and hence improve their comprehension of the book content.

A number of authors have proposed ad-hoc solutions for visually-impaired people. Adjouadi et al. [1] introduced a new automatic book reader for blind people. Their objective is the design of a fully integrated system that is relatively fast, but inexpensive and effective with a high reading accuracy. Kanahori et al. [6] proposed an integrated system for scientific documents including mathematical formulae with speech output interface. "ChattyInfty" is a system composed of a recognizer, editor and speech interface for scientific documents, including PDF documents. In [2] the authors discuss a computer system enabling interactive online presentation of multimedia Daisy books over the Internet. In particular the proposed DaisyReader allows interactive voice reading of math formulas. In our study we prefer to propose solutions to design, develop or adapt accessible and usable e-books, which can be read with traditional tools (e.g. adobe reader, Web browser) in order to obtain more general solutions, with the greatest benefits for all. In [3] the authors discussed an approach for automatically generating an e-textbook on the Web for a user specified topic hierarchy. Their approach consists of generating an e-book on the Web starting with a topic indicated by the user. Users can then browse through the descriptive pages like a book. In [5] the problem related to tagged PDF is faced by using an XML-based structure. The tool implemented is an Acrobat Plug-in. Conversely, our methodology is aimed at developing a semi-automatic support to adapt a single format to various ones, by tacking into account specific user requirements.

3 An Accessible and Usable E-Document

A visually impaired person generally interacts with the operating system and its applications through assistive technology, a screen reader in the case of a blind user and a magnifier for a low vision user. As in the case of Web pages [7] the main problems encountered by a visually impaired person when interacting via these assistive technologies can be summarised as Lack of context, Information overload, Excessive sequencing in reading. Even if a document is available in a format considered

accessible, such as TXT, PDF or HTML, the reader can still find it difficult to read it efficiently. When reading an electronic document users should have at least the same options as when reading a paper document, including being able to obtain an overview of the contents (e.g. sections) and reading each component (e.g. figures, tables or graphs) easily. Therefore, a specific structure, additional navigation features, and appropriate text descriptions should be added to an e-book.

Accessibility is the main feature that makes an e-book readable by everyone and it is not necessarily guaranteed even if the document is available in a digital format. Consideration of usability is also very important since e-books are interactive systems and their access should be easy and consider the needs of their readers. For a document to be accessible, it should have a number of specific features. The main features that make e-documents accessible can be summarised as follows:

- The electronic format should be (e.g. TXT or X/HTML) accessible by a screen reader or magnifier; or the specific reader required for working with a particular format (e.g. DOC or PDF) should be accessible by assistive technologies.
- All figures or graphs should have appropriate alternative descriptions.
- Tables or diagrams should be well-structured so that they are read correctly when sequentialized by the screen reader.

In additio006E to accessibility requirements, usability features are necessary to make an electronic document or book easy to read. In our study we address specific questions related to the usability of the interface of an e-document to be read by blind users. In particular, the main usability features for improving accessible e-documents include methods and techniques to facilitate content and navigation understanding (e.g., title clearness and structure, complex non-textual objects, etc.).

Developing accessible e-documents requires consideration of many different factors. In this perspective we proposed a set of 10 guidelines aimed at improving readability and navigability through screen readers. The proposed guidelines involve three fundamental aspects: (1) the *structure of the document* should be clear. This can be obtained by using appropriate tags for the various type of elements to clearly indicate the main parts of the document (titles, paragraphs, notes, etc.). (2) *Non-textual objects* (such as figures, formulas, tables), if not adequately represented, can create serious access difficulties. *(3) Navigation*, specific mechanisms for navigating in a document should be introduced. For example, if the final format supports links to access pieces of content (such as XHTML) then they should be added. If this is not possible some kind of mechanism to easily identify specific parts (for example, an index at the beginning of the document) should be introduced.

4 Our Approach to Obtaining an Accessible and Usable E-Document

The format of some e-books (e.g. narrative ones) is relatively simple and easily modifiable to improve accessibility. Conversely, in the case of scientific books, such a transformation process is more difficult due to the complexity of the content (e.g. graphs, data tables, etc.). As the development process is generally graphical, and

accessibility and usability rules are rarely considered, such electronic documents need to be modified to make them accessible to readers using assistive technologies. Such a task might be difficult to be carried out by the authors, and it could require a lot of effort as well as a long time. In this perspective, a semi-automatic tool for carrying out the conversion could be very useful.

A number of transformations should be applied to the original format to obtain an accessible version. Such transformations have different levels of automation. For this reason, in addition to the first source format – probably that provided by the publisher – various types of intermediate formats are needed. They can be summarised as follows:

- Source format (Fs), provided by the publisher;
- Pre-intermediate format (Fpi), which is the output of the first semi-automatic transformation;
- Intermediate format (Fi), which includes all needed information (i.e. content, additional descriptions and multimedia objects, etc.) and is the basis for creating the final one;
- Final format (Ff), which is provided to the end user. It may be in TXT, RTF, accessible PDF and DAISY formats.

Regarding the input format we taken into account the PDF format as source document. We based our choice on the fact that any other (not graphical) document can be exported into PDF file. An XML-based solution has been considered for the intermediate formats. The XML-based file was defined in order to contain all needed information (textual and external audio descriptions, content index, and so on). Lastly, for the final output versions several formats are considered beginning from the RTF, TXT, X/HTML as well as tagged PDF, Open Document and daisy formats.

5 The Transformation System

5.1 Main Features and Functionalities

In order to transform an e-document, the person in charge of the transformation has to perform a number of activities, including checking the output of the automatic transformation, processing any parts that need to be corrected, and inserting the alternative descriptions for the figures or the extended descriptions for figures and tables. The editor should enable the possibility of parallel scrolling the original document and the corresponding data extracted in order to allow easy modification. Another possibility is to allow the operator to browse the file scrolling only images, tables and lists without considering the other elements. It also provides the ability to associate text and audio descriptions to images and tables, also allowing the modifications to be saved in the intermediate format. The main features of the environment can be briefly summarised as follows:

- One input format, several output formats. The tool inputs a PDF document and produces several different output formats. In addition to accessible PDF and X/HTML files, the DAISY (Digital Accessible Information System) format is supported.

- Semi-automatic support for transformation activities. The tool performs as many actions as possible automatically, but several activities have to be carried out manually. For instance, descriptions of figures and graphs have to be added manually. Other activities can be carried out by the tool and then the result can be revised by the operator. For example, the tool tries to rebuild the structure by recognising the titles of sections and sub-sections. Using this structure, a table of contents is added to the document and heading styles are assigned. When such activities are carried out automatically, the operators have to check and correct any incorrect titles. The same is true for tables. The tool attempts to rebuild a table from the original version; the operator has to check and probably modify it to ensure the structure is correct.
- Interactive modality to facilitate adapting non-textual objects. As a document can be composed of many pages, the tool offers operators a set of functionalities that assist in carrying out several repetitive activities more quickly. For example, when the operator is adding descriptions to figures and graphs, the tool facilitates the process by showing only the pages that contain images.

5.2 System Architecture

The system is composed of four main components designed to carry out the main phases: (1) the content extractor; (2) the editor; (3) the optimiser; and (4) the final converter.

I. Content extractor: it takes the document and obtains the text and the images trying to rebuild the logical structure through some heuristics. The output is in the intermediate format.
II. Editor: it allows the operator to modify the document in intermediate format generated by the extractor.
III. Optimiser: it takes the result of the modifications made with the editor and generates a document in a valid intermediate format, which applies the guidelines for accessible documents and those for generating the final formats. The resulting valid intermediate format has all the necessary information to generate the final formats.
IV. Final converter: it takes the intermediate format and generates the required final formats (e.g. TXT, RTF, X/HTML and tagged PDF). For this purpose it uses XSLT transformations.

5.3 Content Extractor: The Heuristics

As already mentioned, in our study we assumed that publishers provide their electronic documents in the PDF format. Thus, for the time being, the content extractor takes PDF documents as input. The result of this first automatic processing is an XML-based file, which is then combined with the extracted content as well as all changes and additions made by the operator. The XML-based intermediate format is similar to HTML structure (e.g. H1, H2, etc.) in order to use tags that can be more familiar to the operator. In fact, the automatically extracted content needs to be handled by the operators who can use a graphical editor or a source code editing (see

Figure 1). In addition, the final formats - X/HTML or PDF - should be tagged in order to reproduce the logical structure; thus tags such has <h1>, <p> are used.

In this context, to obtain the document content - text and main non-textual objects - we developed an extraction module using the JPedal library (http://www.jpedal.org) and applying a set of specific heuristics developed to identify rules aiming to better manipulate PDF information and document logical structure. The content extractor first applies one specific heuristic for positioning the figures, whose result is necessary to better apply the other heuristics. The extractor furnishes as much the structural information as possible in order to reduce the manual work required of the operators. The text and formatting styles extracted by the JPedal library are not themselves enough to reproduce a document structure: the extracted data need to be handled and exploited to obtain a logical structure. To this end, we propose a set of heuristics for better rebuilding the content logical structure of the document. The JPedal functions available to handle PDF documents are able to provide the text content, the images and rendering attributes, such as font type, face, size, and the spaces between the words (when there is more than one) indicated with <space count>. These formatting styles (font and spaces) are mainly used to rebuild the document structure. Thus, additional more specific heuristics are needed to obtain more structural information. The main heuristics implemented in the content extractor module are:

Font-size statistic-based reconstruction of paragraphs and headings. A statistical analysis on the font-size is carried out in order to rebuild the content structure (i.e. paragraphs and headings). It assumes that the most often used font is probably associated to the main text, the corresponding text is wrapped in <P> tags. Unfortunately, the JPedal library marks each line as a possible paragraph. Thus, at the end it is necessary to merge some consecutive elements into a single block (see the heuristic below). The biggest font is probably used for the first heading level, thus the corresponding content is marked with <h1>, and so forth. We used <h1>, <h2>, and so on to mark heading levels by analogy with X/HTML structure.

Image positioning. All the images on each page are extracted. To take out a figure from each page is not difficult. Some problems occur when associating the figure to its exact position within the page to be built. This is related to appropriately associate captions with their images as well as to reproduce a well-structured final document with the same visual rendering of the original one. This heuristic identifies the image position by computing the consecutive blank lines. In fact, after conducting some extraction tests, we noticed that the JPedal library leaves a number of contiguous blank lines, which is proportional to the image size. If the consecutive empty lines are sufficient to contain the image then the <img src="..."> tag is included in the target page. The corresponding image is stored in an external file and its path is added to the src attribute. The image is shown when the operator visualises the extracted content by using the graphical editor.

Reconstruction of ordered and unordered lists. To determine the presence of ordered or unordered lists, each textual line is analysed. To identify a potential item, the first character of the line is considered. If the line starts with "1.", it probably means it is the first element of a new ordered list. For the unordered lists, the heuristic checks if the first character is a special symbol (e.g. ".", "*"). Next, the consecutive presence of special symbols in the following lines is verified in order to determine if it could be a list.

Table reconstruction. Based on several conducted tests, we observed that a table is extracted row by row. To recognise that a line is a row in a table, <space count> tags (which indicate a number of consecutive spaces) are searched for: a line containing at least two of such tags is probably a table row.

Merging of adjacent elements. The purpose of this heuristic is to "compact" nearby similar elements into a single block. This is particularly useful to merge consecutive lines recognised as separate paragraphs.

Figure 1 shows the user interface of the extracting module, which, is split into three parts: on the left, the original PDF document; in the centre there are the graphical editor and the resulting document description, on the right, the list of the extracted figures with the possibility to edit corresponding information such as the alternative description or the information for the longdesc attribute. Such a structure should allow operators to access and manipulate both documents: on the left the source document, and then in the remaining parts they can interact with the extracted information. Specifically, the editor allows the operators to work directly on the XML-based descriptions or with information such as all chapter and paragraphs headings.

In the editor, by simply clicking on a figure it is then possible to edit the corresponding alternative description and the longdesc content, and insert an audio explanation. The same functions are also accessible by pressing a shortcut (Ctrl+I). This alternative modality should also allow blind operators to use the editor; they could receive textual or audio descriptions from an expert in the application domain considered in the document, and then proceed to inserting them by using the tool.

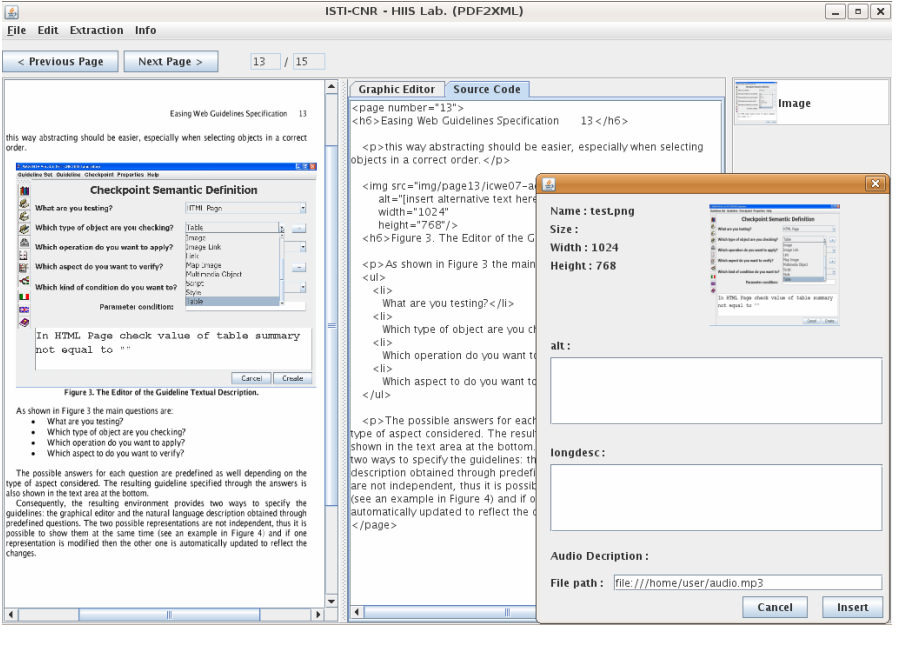

Fig. 1. An example of content extracted from a PDF file

6 Conclusions and Future Work

Making e-documents accessible and usable for vision impaired users is not an easy task. Novel environments are necessary in order to support this activity. In this paper, we have presented a method for supporting such activity and a corresponding prototype. It is based on some automatic transformations that require the support of operators at some points to obtain more meaningful results. We have applied our environment to a first small set of e-documents and the results are encouraging. Future work will be dedicated to engineering the prototype and apply it to a wider set of cases in order to check its general validity.

References

1. Adjouadi, M., Ruiz, E., Wang, L.: Automated Book Reader for Persons with Blindness. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 1094–1101. Springer, Heidelberg (2006)
2. Brzoza, P., Spinczyk, D.: Multimedia Browser for Internet Online Daisy Books. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 1087–1093. Springer, Heidelberg (2006)
3. Chen, G., Li, Q., Jia, W.: Automatically Generating an E-textbook on the Web. World Wide Web 8(4), 377–394 (2005)
4. DAISY Consortium (2005), `http://www.daisy.org`
5. Hardy, M.R.B., Brailsford, D.F., Thomas, P.T.: Creating structured PDF files using XML templates. In: Proceedings of the 2004 ACM symposium on Document Engineering, DocEng 2004 (2004)
6. Kanahori, T., Suzuki, M.: Scientific PDF Document Reader with Simple Interface for Visually Impaired People. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 48–52. Springer, Heidelberg (2006)
7. Leporini, B., Paternò, F.: Increasing Usability when Interacting through Screen Readers. Springer International Journal Universal Access in the Information Society (UAIS) 3(1), 57–70 (2004)
8. Sun, Y., Harper, D.J., Watt, S.N.K.: Design of an e-book user interface and visualizations to support reading for comprehension (Posters). In: Proc. of ACM SIGIR 2004, pp. 510–511 (2004)