[MZBRSt07] *Michael Metzger, Birgit Zimmermann, Sonja Bergsträßer, Christoph Rensing, Ralf Steinmetz;* **Automating Layout Adaptation of Textual-based E-Learning Content;** World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA 2007), Vancouver, Juni 2007

Automating Layout Adaptation of Textual-based E-Learning Content

Michael Metzger[2], Birgit Zimmermann[1, 2], Sonja Bergsträßer[2], Christoph Rensing[2], Ralf Steinmetz[2]

[1] SAP AG
SAP Research CEC Darmstadt
Bleichstr. 8, 64283 Darmstadt, Germany
birgit.zimmermann@sap.com

[2] KOM Multimedia Communications Lab
Technische Universität Darmstadt
Merckstr. 25, 64283 Darmstadt, Germany
MichaelMetzger@gmx.net
{birgit.zimmermann, sonja.bergstraesser, christoph.rensing, ralf.steinmetz}@kom.tu-darmstadt.de

Abstract: The reuse of existing content plays a significant role in the increasing market of E-Learning content. But often this reuse of content requires adaptations to adjust the content for its new usage. This paper focuses on adapting the layout of E-Learning material. Based on an analysis of layout adaptations in textual-based E-Learning formats a concept and a prototype for a (semi-) automatic layout adaptation tool are presented.

1. Introduction

The reuse of existing material helps to produce high-quality E-Learning content faster and less expensive. But due to changed usage scenarios reusing E-Learning content often requires adaptations. In (Zimmermann et al. 06a) a survey of requirements on adaptations of reusable (E-Learning) content is given. Here 15 different kinds of adaptations (e.g. adaptation to a changed learning objective or adaptation to another language) are presented. The adaptation that is performed most frequently is the adaptation to a changed layout. Performing this layout adaptation manually needs time and skilled workers. Whereas automating it would save time and costs.

In this paper a concept for automating layout adaptations is presented. First an overview of the challenges of layout adaptations is given. In chapter 3 the outcome of a theoretical analysis focusing on often used textual based E-Learning formats is presented. Chapter 4 discusses related work. Based on the outcome of the survey the design of a layout adaptation tool is explained in chapter 5. Chapter 6 contains a summary and an outlook on future work.

2. Challenges of Layout Adaptation

E-Learning content has two characteristics: First, there are many formats used. Often even one course contains several formats, e.g. a course in HTML may contain flash animations. In addition often specific files are used with formatting instructions, like CSS for HTML. When adapting the layout of a course all relevant files of a course have to be considered. Second, a typical E-Learning course is composed of many files. Consequently, layout adaptation of E-Learning content has to handle both: many different formats and lots of files.

In the following the objects of a layout adaptation are called *layout elements*. A layout element is an enclosed object of a course layout – for example headings or logos. (Image 1 shows some layout elements.) These layout elements have to be detected and adapted differently in different formats. Furthermore some layout elements are only supported by some formats.

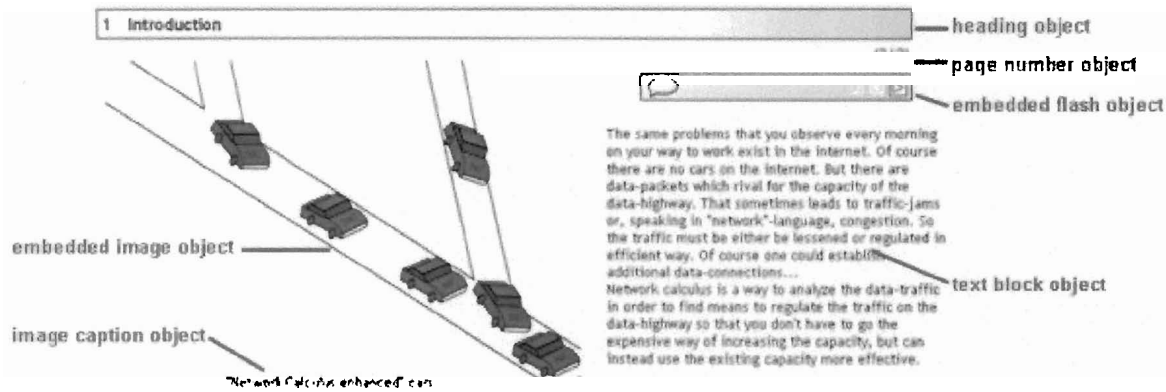

Image 1: Layout elements of an example E-Learning course.

Often the same adaptation has to be done several times when adapting the layout of one course, e.g. changing a logo on each page of a course. It is very inefficient to handle this manually. Therefore it is desirable to automate layout adaptation or, where full automation is not possible, to assist human workers with a semi-automatic adaptation tool.

Automatic layout adaptation is composed of four steps: 1st Reading the course data, 2nd Detecting layout elements, 3rd Adapting them, and 4th Write the adapted course. Detecting the elements is the most complicated issue, as often many possibilities exist to encode the same layout element. This will be discussed in detail in chapter 4 and 5.

Regarding the arrangement of layout elements two problems occur when executing layout adaptations: First the overlapping of elements: In some formats resizing or adding elements might cause overlapping with other elements. And second displacements of layout elements: They might appear, if resizing, adding, or deleting of elements lead to a re-arrangement of the surrounding elements. In some cases the new arrangement does not comply with certain style rules and therefore has to be avoided.

3. Study on Layout Adaptations

To understand the format specific characteristics of layout adaptations, several formats, which are often used in the context of E-Learning, have been analyzed in respect of their format specific characteristics. In addition the adaptable layout elements occurring in these formats have been determined. To get a good overview on the characteristics of the formats the following common formats have been chosen for the study:

- The Microsoft Word and Oasis OpenOffice (http://www.openoffice.org/) Write format
- Adobe Acrobat PDF
- The Microsoft Powerpoint and OasisOpen Office Impress format
- HTML and CSS, XML (with CSS/XSL/XSLT for browser output)
- Lecturnity (http://www.lecturnity.de/) and Smil2.0 (http://www.w3.org/TR/2005/REC-SMIL2-20050107/)

The study was divided into two parts: Studying the documentations of the formats and analyzing a set of existing E-Learning courses. Based on the results of the study different groups of layout elements have been identified, like captions (headings, image labels, and table labels, etc.) or elements that occur with a fix position on every page (author, copyright labels, creation date, or page numbers). In sum 10 different kinds of elements were found.

Regarding possibilities to detect all these elements strong differences between the different formats were found: XML-based formats mostly have a clear element labeling (e.g. by using special tags). Therefore it is easy to detect those layout elements. The opposite are formats like PDF, in which characteristics of elements (e.g. font size) normally are hard coded. No element labeling for direct element identification is available. In addition often element labeling exists, but is not used as intended (e.g. heading tags used for image captions). This is primarily a problem of

HTML. For example over 90% of actual existing German HTML-sides are malformed. (ValiWatch 05) This complicates the design of a format comprehensive automatic layout adaptation tool.

4. Related Work

4.1 Authoring tools

This subchapter focuses on Authoring tools for HTML and CSS. In HTML and CSS the number of adaptation tools is nearly countless. Why not use one of these existing tools for adaptations? To answer this question, an analysis of many different existing HTML authoring and adaptation tools has been carried out. In the following two examples of typical HTML/CSS authoring tools will be examined to explain why it is necessary to design a new tool.

The first tool is the freeware CSS-authoring tool *Simple CSS* (http://www.hostm.com/css/). Simple CSS is a small tool to adapt CSS files. It offers a graphical interface to users, in which they can set many characteristics of the different CSS-rules via drop-down menus (e.g. color, size, etc.). In well-encoded HTML courses, with a clear separation between content (in HTML files) and layout (in CSS files) and a known namespace for CSS rules, Simple CSS is a good assistant for layout adaptation. But these conditions are not met by most E-Learning courses. In addition the tool only allows users to adapt each characteristic of each CSS rule manually.

A second tool is the widely-used HTML authoring tool *phase 5* (http://www.qhaut.de/). It is based on a simple source code editor, which is expanded with many functions to generate and adapt html code. phase 5 is able to handle all kinds of HTML code including malformed code. But as in Simple CSS users have to adapt every file separately. In addition users need good HTML skills to work with phase 5.

Many tools exist for changing the layout of HTML material. But at the moment no tool is known, which meets all requirements for automated layout adaptation or at least most of them. Therefore it is necessary to design a new tool.

4.2 Element detection

As explained before, good detection algorithms are needed for automatic layout adaptation. At the moment no approach exists dealing with detection and adaptation of all layout elements in order to adapt the layout to a changed (corporate) design. In this chapter existing approaches for logo detection and topic segmentation are presented.

Research in the area of automatic page layout adaptation of HTML sites for small form factor devices has a short history but is growing fast. Most approaches in this field focus on partitioning large web pages in expedient semantic blocks. Layout elements are useful indicators for possible separators, but concrete approaches to detect such elements are still very rare. (Cai 04, Bickmore et al. 97)

In addition, there are many approaches for automatic logo and trademark extraction/removal on websites. Baratis (Baratis 05) presents an approach for logo detection in corporate web-sites based on a histogram method. He extracts pictures from a website and analyses their color histogram as well as their radial-frequency-histogram after a Fourier-transformation. Logos are detected by special characteristics (few colors, clear separation between colors). Other approaches deal with extractions of logos from videos. Kovar (Kovar 02) presents an approach for the detection of logos in sport videos which is based on the detection of sharp edges in single video frames. Yan (Yan 04) and Boujemaa (Boujemaa 04) demonstrate an algorithm which is based on the detection of elements that are not modified in successive frames.

Another interesting point is topic segmentation. Topic segmentation is motivated by the fact that often the text of a HTML page contains more than one topic without clear labeling for separation. As topics can be used to detect text blocks or headings, topic segmentation is interesting for the detection of layout elements, too. Reynar (Reynar 98) describes inter alia an algorithm to identify topic boundaries, which is based on a semantic content analysis.

In conclusion there are some approaches, which deal with element extraction in text formats. Often they aim on the segmentation of large web sides in order to adapt them to small factor devices or to generate indexes or metadata for

the pages. These approaches provide interesting techniques, which can be used for automatic layout element detection. The research on detection possibilities in image or video formats is not directly applicable to text based formats. But underlying ideas are in some cases useful for text based formats, too. (E.g. the idea that typically logos appear at a similar position in successive frames also applies to slides or pages of an E-Learning course.)

4.3 Formats designed for reuse and adaptation

There exist a few formats which are designed for easy adaptation and reuse. In the following the widely-used format SMIL and the ZYX approach are described as examples for those formats.

SMIL (SMIL 2005) is a XML based language that allows the integration of many different multimedia formats. It has a clear separation between content and structure. It is possible to describe the behaviour of the different multimedia objects and the layout of the presentation. SMIL is one of the first presentation formats, which allows easy adaptations to different purposes or presentation devices. But SMIL is designed for dealing with SMIL content not with other formats. In addition it does not allow layout adaptations in every detail.

ZYX (Boll et al. 01) is a hypermedia format that allows the adaptation and exchange of single content-fragments from a course. The goal of ZYX is to facilitate the reuse and adaptation of multimedia content. The ZYX-model realises this by a tree structure, in which every node represents a presentation element. Various characteristics describe the behaviour of an element and its descendants. But ZYX is also not suited for all kinds of content.

5. Prototype for a (Semi-) Automatic Layout Adaptation Tool for HTML

5.1 Concept

So far the need of an automatic support for layout adaptation has been shown. A tool that is suited to help users in performing a layout adaptation for HTML courses at least semi-automated has to fulfil the following four demands:

1. Users of the adaptation tool should not need special knowledge in different E-Learning formats, as they are normally subject matter experts but not experts in dealing with all formats. This should be realized by a user interface, which abstracts from the format of the content, and which is intuitional to run.
2. The adaptation tool has to handle courses with many files. This affects the runtime behaviour of the tool. As the detection algorithms are the bottleneck of the performance they have to be designed efficiently.
3. In addition, it should be possible to adapt different courses according to a reusable template of adaptation instructions. Often many courses or parts of a course have to be adapted to the same design guideline. Therefore it should be possible to provide an adaptation template that is used for several adaptations.
4. The tool has to be able to handle malformed HTML and CSS.

Cai (Cai 04) divides detection algorithms into two categories: Syntactic techniques operating on the structure of a web space (or in this case an E-Learning course), and semantic techniques focusing on understanding of meaning. A (semi-)automatic layout adaptation tool has to use both. Syntactic techniques – for example based on labels or tags – are good approaches to detect layout elements. But as mentioned before HTML tags are not always meaningful. So there is a need for a semantic detection, too. The theoretical analysis has shown, that automating the detection based on semantic techniques is difficult and not always possible. Therefore the layout adaptation tool presented here uses semantic heuristic techniques as well as supporting user interaction in cases where the semantic detection does not deliver a useful result. This helps to achieve a maximum of reliable detection with a minimum of user interaction.

5.2 Implementation of a prototype for the HTML format

To prove the outcomes of the study a prototype has been implemented. The prototype is written in Java and uses the Cyberneko HTML Parser (http://people.apache.org/~andyc/neko/doc/index.html) to parse the HTML files of a course. NekoHTML is a tag balancing parser which corrects many mistakes in malformed HTML code. To handle the HTML data JDOM (http://www.jdom.org/) is used. The necessary CSS data is read by a simple open source CSS

parser called "cssParser". The cssParser API (http://cssparser.sourceforge.net/) provides objects to handle the parsed CSS data. The graphical user interface of the prototype is based on Java swing.

There are five groups of implemented layout adaptation algorithms:

- Adaptation of document-background
- Adaptation of headings
- Adaptation of images (exchange, remove)
- Adaptation of tables (whole table, table header, table row, table cell)
- Adaptation of unsorted lists (whole list, single list levels)

The element detection algorithms between the single element groups are completely different. But they are all based on a three-step concept:

- First, the tool tries to detect a desired layout element by characteristic tags and asks the user for correctness
- If the user does not accept the result, the tool tries to detect the element by heuristic techniques
- If heuristic techniques also do not provide a satisfying result, the tool offers the possibility of a detection based on user interaction. There are many different possibilities of interaction based detection algorithms implemented, for example the detection of headings based on the input of a snippet from one example text of the desired heading class. The tool then analyses the whole course and presents all headings, which have the same structure as the one determined by the user. These algorithms deliver very good results.

Not every implemented algorithm includes all of these three steps. Some groups of layout elements can not be detected by typical tags. In this case the first step is not executed. For other groups it is difficult to create a good heuristic for detection. In other cases steps two and three are not required.

5.3 Evaluation

The prototype is able to handle many layout element adaptations in HTML. The provided adaptations cover a large area of possible layout adaptations. Image 2 shows screenshots from one slide of an E-Learning course – before and after the adaptation - that illustrate some possibilities the prototype offers.

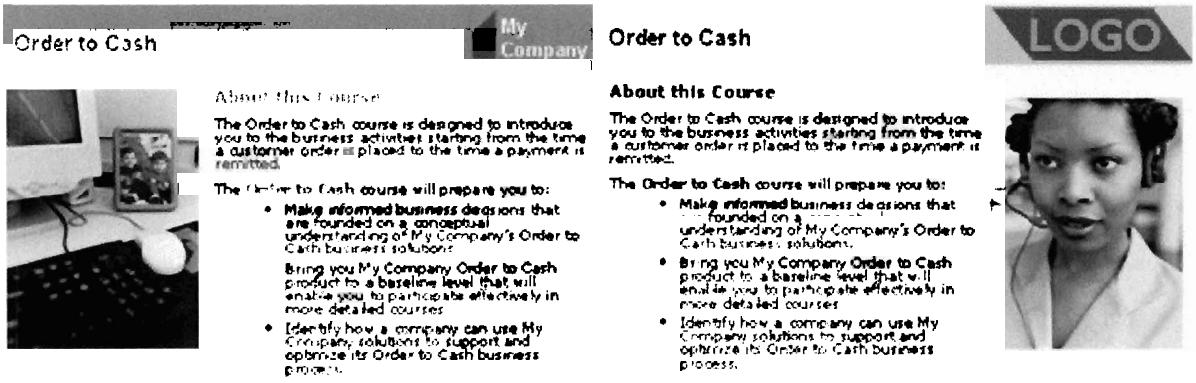

Image 2: Example E-Learning Course before and after Adaptation.

Many of the implemented algorithms can be used for similar adaptations (e.g. the detection and adaptation of images are similar to the detection and adaptation of other objects, like embedded videos). Even difficult challenges are solved with small and easy to handle user interactions (e.g. to detect headings, which are labelled with not expected tags or the identification of images). But there are some unresolved problems like the detection of elements in very malformed code and multiple nested tags. In addition the prototype is restricted on HTML and CSS, but often E-Learning courses are using java script to define special layout elements. Those problems have to be solved in future projects. But the even the restricted prototype shows some really promising results.

6. Summary and Future Prospects

This paper gives an overview of an analysis of problems occurring in automated layout adaptations. In addition a prototype of a layout adaptation tool for HTML is presented. The prototype shows possibilities for dealing with difficulties in adapting HTML (e.g. the heading detection by analyzing a given example). The implemented prototype is a first step for a general layout adaptation tool. Many principles of this prototype are applicable to other formats. In particular other XML based formats like OpenDocument or the Microsoft Office XML format could be treated in a similar way.

Based on the experiences of the implementation of the presented prototype the next step is to design a comprehensive layout adaptation tool, which can deal with most relevant e-learning formats. It is recommendable to design this as one tool, which is based on a large set of basis algorithms and not as separated tools for all formats, as often many formats are used in one course and then one tool is more user-friendly than many separated tools.

7. Acknowledgements

This work is supported by the German Federal Ministry of Economics and Technology in the context of the project Content Sharing.

8. References

Baratis E. (2005): *Automatic Logo and Trademark Extraction from Large Corporate Web-Sites*, Technical Univ. of Crete (TUC), Dept. of Electronic and Computer Engineering, Chania, Crete, Greece

Bickmore, T.W., Schilit, B.N. (1997), *Digestor: "Device-independent Access to the World Wide Web"*, Computer Networks and ISDN Systems, vol. 29, nos. 8-13

Boll S., Klas W. (2001): *ZYX - A Multimedia Document Model for Reuse and Adaptation*, In: IEEE Transactions on Knowledge and Data Engineering 13(3), IEEE Computer Society

Boujemaa, N., Fleuret, F., Gouet, V. and Sahbi, H. (2004): *Automatic Textual Annotation Of Video News Based On Semantic Visual Object Extraction*, IS&T/SPIE Conference on Storage and Retrieval Methods and Applications for Multimedia, San Jose CA, USA

Cai, J. (2004): *Page Layout Adaptation for Small Form Factor Devices*
http://www.cs.toronto.edu/~jcai/2514/term.pdf (Last accessed 01.10.1006)

Kovar, B. (2002): *Logo Appearance Detection and Classification in a Sport Video*, 3rd COST #276 Workshop, Budapest, Hungary

Reynar, J. (1998): *Topic segmentation: Algorithms and applications*, In PhD thesis, University of Pennsylvania

SMIL 2.1, W3C Recommendation (2005), *Synchronized Multimedia Integration Language (SMIL 2.1)*,
http://www.w3.org/TR/2005/REC-SMIL2-20051213/smil21.html (Last accessed 01.10.1006)

ValiWatch (2005) - *Untersuchung zum deutschsprachigen Web*, validome.org, 2005:
http://www.validome.org/lang/ge/html/valiwatch-web-2005 (Last accessed 01.10.1006)

Yan, W., Wang, J., Kankanhalli, M. (2004): *Automatic video logo detection and removal*, IZ FH-Potsdam

Zimmermann, B., Bergsträßer, S., Rensing, R., Steinmetz, R. (2006a): *A Requirements Analysis of Adaptations of Re-Usable (E-Learning) Content*, AACE EdMedia