Content Exposure of Slide Show Presentations for Selective Download and Annotation via Mobile Devices

Philip West, Greg G. Foster, Peter G. Clayton
Department of Computer Science
Rhodes University
Grahamstown, South Africa
g00w4340@campus.ru.ac.za, g.foster@ru.ac.za, p.clayton@ru.ac.za

ABSTRACT

Presentation software, such as Microsoft PowerPoint, has gained popularity as a method of lecture delivery, and the built-in web publishing features make it easy to convert such slide shows to HTML format. The increase in the availability and functionality of handheld devices has provided new opportunities for using mobile technology in education. We propose the novel integration of these technologies for use in a teaching environment that allows users to selectively download and annotate parts of the slide show using mobile devices. A web application has been developed for use in a lecture theatre or classroom environment that allows students to access content from a slide show during a presentation. An HTML version of a PowerPoint slide show is made available via a wireless access point. A server-side application parses the slides contents to create an image representation of the various content groups. A mobile web application displays these content groups as different coloured blocks which correspond to the content on the main lecture display screen. Each coloured block provides a link to selectively download the desired content group, which the server application then extracts from the slide's contents. In addition, the mobile application allows for the annotation (lecture notes) and storage of the downloaded data. Students can then view the downloaded data and annotations at a later stage for study purposes.

Keywords: Digital Slide Show, Content Exposure Annotated Content, Spatial Representation.

This work was undertaken in the Distributed Multimedia Centre of Excellence at Rhodes University, with financial support from Telkom SA, Business Connexion, Comverse, Verso Technologies, THRIP, the National Research Foundation, and Microsoft.

All authors are with the Computer Science Department, P O Box 94, Rhodes University, Grahamstown 6140, South Africa.
(P. West. g00w4340@campus.ru.ac.za; 046 603-8642)

1. INTRODUCTION

Rapid advances in mobile device technology have resulted in an increased adoption of these devices in the corporate sector and, more recently, in the educational sector (Finn and Vandenham, 2004). In addition, mobile technology has opened up new opportunities to create novel learning experiences (Rogers *et al.*, 2004). While traditional computer laboratories offer occasional exposure to learning technology, mobile devices now provide the opportunity for frequent exposure making them an integral part of the learning process (Roschelle, 2003). Mobile devices have also been shown to significantly increase students' ability to manage their learning environment and to collect information (Zurita and Nussbaum, 2004).

Formal lecture sessions continue to be a popular method of presenting information to university students, and, over the last ten years, digital slide show presentations have become the standard medium of delivery (Bartsch and Cobern, 2003). Students are often encouraged to make study notes during lectures. They then may or may not have access to the original presentation slides after the lecture. Common presentation software packages, such as Microsoft PowerPoint, have built-in web publishing features which make it easy to convert slide shows to HTML format. This, together with numerous content extraction approaches for web content (Alam and Rahman, 2003; Buyukkokten *et al.*, 2001; Chen *et al.*, 2003; Han *et al.*, 2000; Kunze and Rosner, 2001; Gupta *et al.*, 2003; Gupta *et al.*, 2004), and ubiquitous mobile computing has provided an opportunity for the novel integration of these technologies for use in the teaching environment.

We present a system that applies content extraction techniques to digital slide show presentations, and provides a mechanism for annotating selected content and for storing these notes. This system illustrates the use of mobile devices as a support tool in traditional learning environments. The main role of the system is therefore to augment learning, and not to deliver instructional content.

2. SLIDE SHOW CONTENT EXPOSURE

The standard architecture for delivering lecture presentations incorporates a large display screen, whereby the displayed content is controlled by a desktop or laptop computer (Figure 1a). Our content exposure system extends this approach by introducing a server coupled with a wireless access point (AP) (Figure 1b). The server hosts the content exposure system, implemented as a mobile web application, and the actual presentation content in HTML format. Mobile device users connect to the server through the wireless AP and are provided access to the lecture content through the exposure system (Figure 1b).

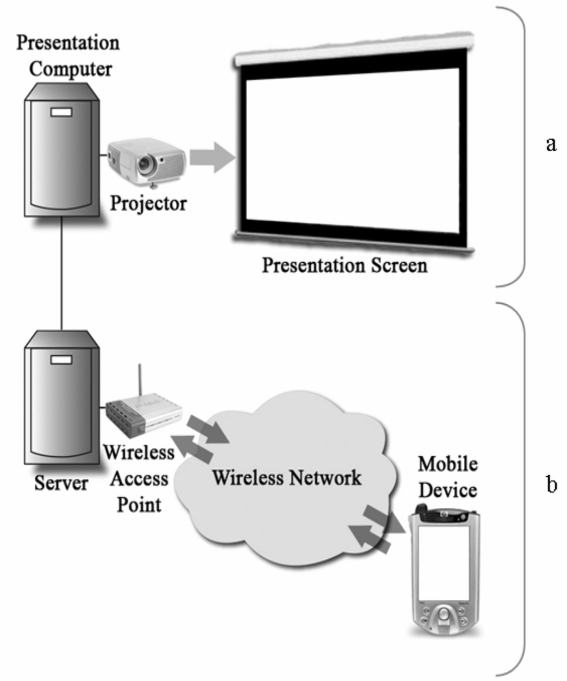


Figure 1. High-level architectural view of the slideshow content exposure system. (a) The standard lecture presentation architecture. (b) An extension to the standard architecture providing the content extraction system and wireless mobile device access.

In order to apply standard content extraction techniques it is necessary to convert slide show files into a markup language. Slide show presentation software, such as Open Office, converts each slide into a single image which is then displayed in an HTML document. This makes identifying content within a slide complex since it would be necessary to analyze the image. However, other presentation software, such as Microsoft PowerPoint, maintains the content from the original slide during the conversion process and represents each content object separately, thus making content extraction a simpler process. In addition, PowerPoint is supplied with an abstract programming interface (API) for use with the .NET framework, which exposes certain functionality and allows certain processes to be automated. PowerPoint was therefore the presentation software selected as a base format for creating and storing digital slide show presentations for use with the content exposure system.

2.1. PowerPoint Conversion

Conversion from PowerPoint file format to HTML is automated by utilizing the PowerPoint API in a C# windows application. Conversion is carried out in code by loading the desired PowerPoint file, then executing the “Save As” command and specifying HTML version 3 as the output type. The C# application will search a specified directory for previously unconverted PowerPoint files at a set time interval, and automatically convert any appropriate files discovered. After conversion the necessary content extraction techniques are applied to the generated HTML file.

2.2. Content Extraction

Content extraction comprises a collection of techniques that have previously been used in an attempt to display normal web content on small screen devices such as PDAs. Documents are therefore manipulated to be more suited for display on handheld devices.

One of the techniques, referred to as text summarization, improves the network access times of handheld device users and reduces the number of required pen actions while browsing the internet (Buyukkokten *et al.*, 2001). This approach displays web pages in text, and allows sections of the pages to be hidden, partially displayed, summarised or displayed in full.

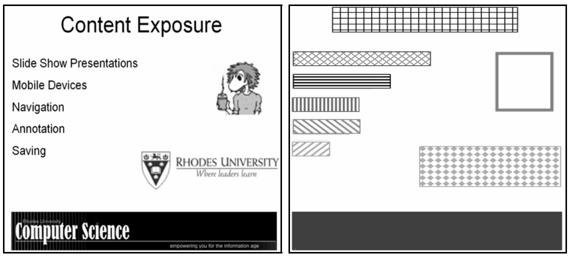
Text summarization has been extended to include a solution for handling graphics by Chen *et al.* (2003). A web page is arranged into a two-tier hierarchy. The first tier provides an overview of the web page in the form of a graphical table of contents, which is displayed as a thumbnail image (Figure 2a). Content is grouped according to semantic relation, and these groupings are then displayed on the thumbnail in different colours. In order to view content the user selects a colour block, which has been formatted for display on a small screen (Figure 2b). There are two main processes used to identify content blocks: page analysis and page splitting or auto positioning. The web page is first parsed into a Document Object Model (DOM) tree to allow for manipulation. Page analysis attempts to extract the semantic structure of the web page, grouping content by inferring the content layout. This is derived from high-level content blocks, such as headers, footers and sidebars, and visual separators such as white space.

(a) (b)

Figure 2. Text summarization extended to include graphics (Buyukkokten *et al.*, 2001). (a) A web page represented as a thumbnail image. (b) The content corresponding to the block selected.

Single-subject splitting divides the page into several smaller pages which are connected by hyperlinks. This technique is used for pages with a particular topic. Multi-subject splitting divides the page into smaller pages, and generates an index page which is used for navigation. This approach has been used to split an entire web site. Auto-positioning is suggested as an alternative to page splitting. The process is similar in that a thumbnail view of the page is generated, displaying the content blocks in different colours. However, when a particular block is selected, instead of splitting the page it is automatically positioned correctly to display the selected content. Certain implementations of this technique have extended this approach to split pages and deliver the sections to different devices (Han *et al.*, 2000).

We have developed a spatial representation algorithm, based on the work of Chen *et al.* (2003), which converts digital slides into a graphical representation. The process creates an image representation of each slide, where each content group is displayed as a differently coloured block. In order to achieve this, the HTML is converted into a document object model (DOM) tree. This tree is then rendered using the ICEBrowser SDK, which is a Java web browser library (ICESoft, 2005). Once rendered, bound box coordinates for each content block can then be extracted. A spatial representation image is then generated (Figure 3b), representing each content block from the original slide (Figure 3a), by using the bound box coordinates previously obtained.


Content Exposure

- Slide Show Presentations
- Mobile Devices
- Navigation
- Annotation
- Saving

(a) (b)

Figure 3. Results of the spatial representation process. Coloured blocks have been represented here as various hatch fills for clarity. (a) The original slide. (b) The resulting spatial representation image.

2.3. Text Summarization of Slides

An XML file is also created during the spatial representation process, and this is used to store a detailed description of each slide in the presentation, including: bound box coordinates for each content block, the names and locations of all images and the actual HTML content for each block. The bound box coordinates are necessary to create the image map using the image generated during the spatial representation process. When a user clicks on an area of the image the screen coordinates are captured and are matched against the bound box coordinates stored in the XML file, locating the corresponding content block.

2.4. Front-end Application – ASP.NET Mobile Web Application

The client-side interface was implemented as a Microsoft ASP.NET mobile web application, which has the benefit of requiring no smart client application, only a web browser. A further advantage is that the majority of the application processing is carried out on the server-side, reducing the load for the limited processing power of the mobile device (Hadjisotiriou *et al*., 2004).

The initial application screen provides the user with mechanisms to navigate through the slideshow or to navigate within a single slide (Figure 4). The "Next" and "Previous" buttons allow for navigation to the desired slide. A label indicates the index of the current slide, and corresponds to the actual slide being shown on the large display screen (which also bears an index number). This assists users to follow the slide show presentation and to navigate to the currently displayed slide. Information that is desired can be selected, and the content is displayed in the annotation screen (Figure 5). Annotations can be added using the textbox. The "Add" button temporarily stores annotations and the corresponding content on the server before returning the user to the previous screen. The "Cancel" button will return the user to the active slide without storing annotations or content. Generally, selecting a content block from the spatial representation image map of the slide will navigate the user to the annotation screen (Figure 6a). However, it is also possible to view and annotate multiple content blocks by checking the "Multiple" checkbox. This prevents navigation to the annotation screen until the user selects the "Annotate Selected" button (Figure 6b). The "Save" button (Figure 4) prompts the user to save all annotations and selected content. Notes are saved as a Rich Text Format (RTF) file. RTF was developed as a format that would be readable by all common word processing applications, and can be viewed on most operating systems. RTF is capable of storing images and basic formatting information. Macros can not be inserted into RTF files, and the format is thus not susceptible to macro viruses. RTF files require less memory than other word processing formats, making the format useful for web downloads and desirable for mobile devices owing to their relatively small memory capacity (Senior Corps Tech Centre, 2005).

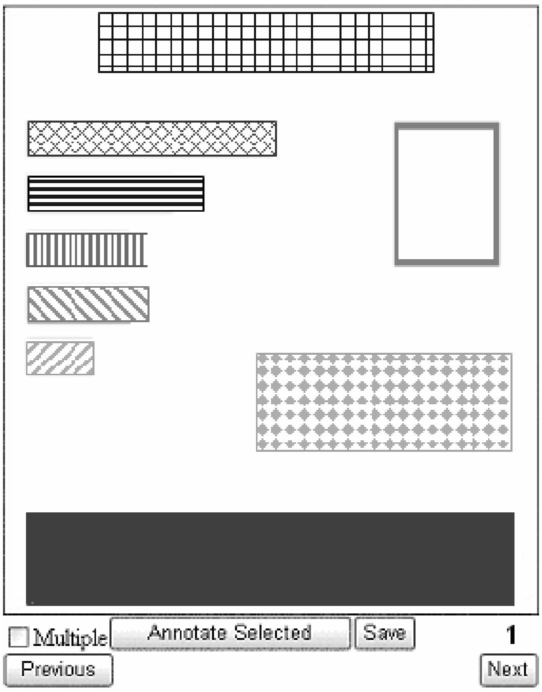


Figure 4. The web application user interface.

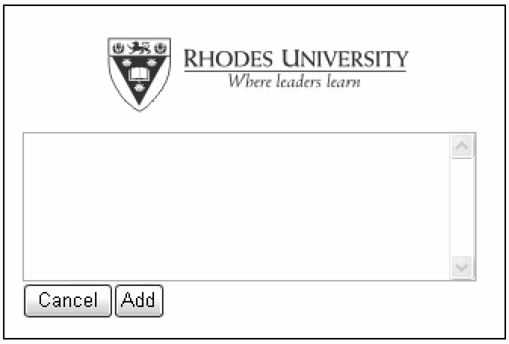
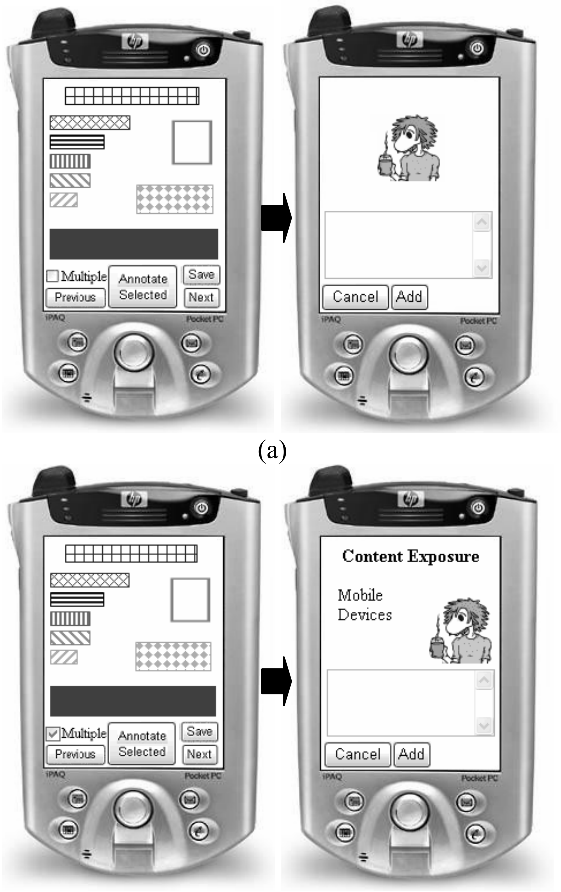


Figure 5. The annotation screen of the web application, showing an image to be annotated.



(a)

(b)

Figure 6. Using the image map to display selected content. The representation corresponds to the slide shown in Figure 3a. (a) The results of selecting a single content block. (b) The results of using multiple select to select numerous content blocks.

3. IMPLEMENTATION TESTING

3.1. Complexity

PowerPoint slideshows, and indeed individual slides, can vary greatly in their levels of complexity. In order to determine the robustness of our system and to observe how the system would handle slide presentations of varying complexity, we categorized slides into two groups: simple and complex. Simple slides contain only static objects, and are limited to certain types of objects such as text boxes, images and tables. Alternatively, complex slides contain animated objects and can consist of an unlimited variety of objects. A complex slide was tested using the spatial representation system. On conversion from PowerPoint format to HTML, it was discovered that PowerPoint automatically removes any imbedded custom animations thus removing the possibility of creating any artifacts. Figure 7 (a - d) illustrates example custom animations, where an image is added after each mouse click. Converting this slide to HTML will produce only the view shown in Figure 7d. Thus conversion removes underlying slide complexities, eliminating the need for modifications to the spatial representation algorithm. The resulting representation is shown in

Figure 7e. Hence, the system is capable of handling complex slide shows without reducing the accuracy of the representations.

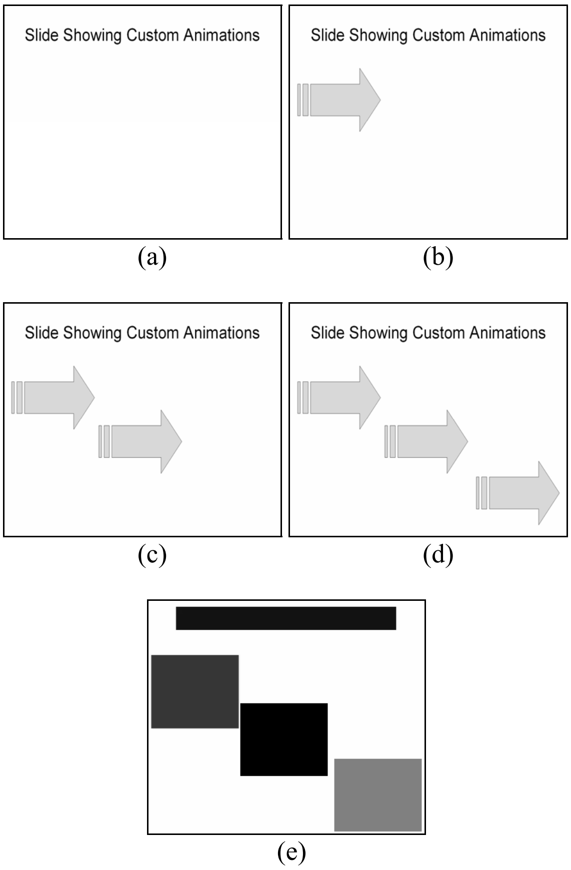

Slide Showing Custom Animations
Slide Showing Custom Animations
(a)
(b)
Slide Showing Custom Animations
Slide Showing Custom Animations
(c)
(d)
(e)

Figure 7. Slide with complex features. In this case custom animations. (a) Initial view. (b) First animation, an image is added. (c) Second animation, another image is added. (d) Third animation, the final image is added. This view is also the result of converting this slide to HTML format. (e) The resulting spatial representation image.

3.2. Adaptation to Screen Size

Mobile devices have varying screen dimensions. The front end system provides the ability to adapt to this difference in device specification by using the device filtering approach implemented in ASP.NET (Homer *et al*., 2005). Device filtering allows criteria to be set depending on the requesting device. We implement a filter which delivers the appropriately sized spatial representation image depending on the device screen size, allowing the system to remove the need for users to scroll on the navigation page.

3.3. Load Testing

Since the system is designed for use in classroom and lecture situations, it was imperative to evaluate the maximum number of possible users that the system can accommodate. Testing was carried out in a controlled environment using the testing facilities provided by Microsoft Visual Studio. The server used was a Pentium 4, 2.80GHz CPU with 1.00GB of RAM, running Microsoft Windows Server 2003 Standard Edition. A web test was created that simulated a web session where a user accessed a presentation using the mobile web application. Actions included navigation, annotation of content, and saving the selected content. A load test was then created to evaluate the load handling capabilities of the web application. The load test began with 1 user, and increased the number of users by 10 every 10 seconds until server errors were encountered. Each simulated user carried out the web test created previously. The amount of available bandwidth was limited to an even distribution between 1.5 Megabits per second (Mbps) and 6Mbps in order to simulate an environment similar to that of a wireless network with minimal bandwidth (Varshney, 2003). The test browser was Pocket Internet Explorer 3.02. User "think times" were centered on a normal distribution using the user think time from the web test. Results from the load test indicate that the web application is capable of handling approximately 118 simultaneous connections transmitting about 35 requests per second. As user load (---, Figure 8) increases, the number of requests per second increases until the maximum is reached and test cases begin to fail (- - - , Figure 8). The average time to last byte (TTLB) (Figure 9) gives an indication of user wait time. After the first few connections have been processed, the TTLB stabilizes at approximately 0.025 seconds. This illustrates how ASP.NET caches data, benefiting users by reducing the response time of the system. The number of requests per second from the test case (35) is much higher than would be expected in a lecture environment where requests per second for each student would be significantly lower. Thus we predict that the system would be capable of handling a large student class size.

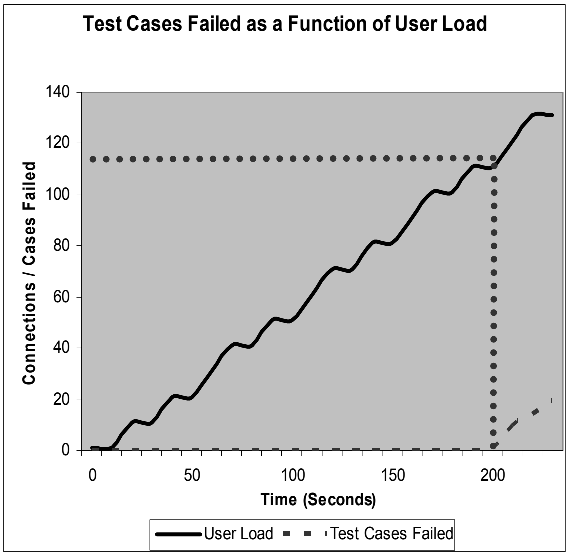


Figure 8. Results of a load test carried out on the mobile web application showing the user load and the number of failed test cases.

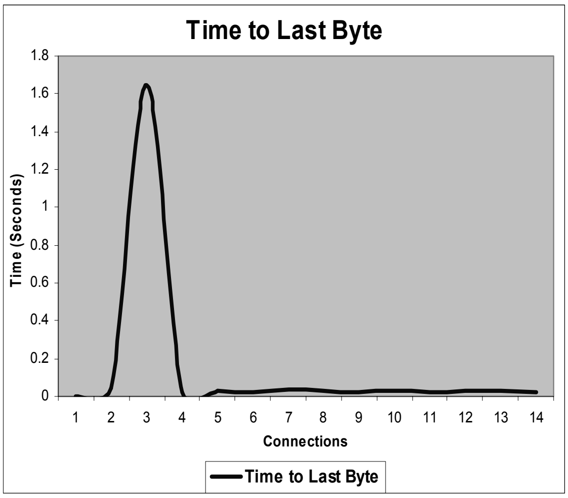


Figure 9. Results of a load test carried out on the mobile web application showing the average Time to Last Byte.

3.4. Field Trials

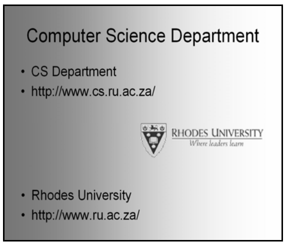
A trial presentation was conducted in order to test the system in a real-world environment. Seven users, who had not previously used the system, were equipped with a variety of mobile device (2 laptops and 5 PDA's) and utilized the system during a simulated lecture session lasting approximately 20 minutes. All aspects of the system were tested including navigation, annotation and saving. It is important to note that the system functionality was evaluated, and not the influence of the system on the learning process. Each user was required to complete a questionnaire at the end of the session. Results from this preliminary investigation into user acceptance of the system were favorable (Table 1). The mobile web application functioned correctly on both laptops and PDAs. The system is simple, allowing users to quickly learn how to use it. Users were able to use the system effectively during a presentation. Users found the spatial representation intuitive, and it allowed them to effectively select the correct content. Overall, users were positive about the system and considered it to be a productive tool.

Questions	
1. Overall, I am satisfied with how easy it is to use this system	1 2 3 4 5 **6** 7
2. I can effectively complete my work using this system	1 2 3 4 **5** 6 7
3. It was easy to learn how to use the system	1 2 3 4 **5** 6 7
4. I believe I became productive quickly using this system	1 2 3 4 5 **6** 7
5. This system has all the functions and capabilities I expect it to have	1 2 3 **4** 5 6 7
6. I found the spatial representation of the slides to be intuitive	1 2 3 4 **5** 6 7
7. I could easily keep up with the presenter	1 2 3 4 **5** 6 7

Table 1. Average results of a field trial evaluation of the content extraction system (1 corresponds to a strong disagreement, while 7 corresponds to a strong agreement).

4. LIMITATIONS

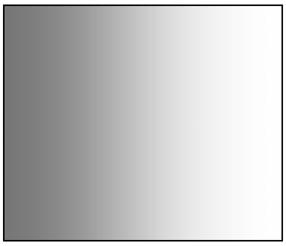
Although the spatial representation algorithm accurately analyses slides, the conversion process from PowerPoint files to HTML may produce some inaccuracies. For instance, the slide shown in Figure 10a consists of various content objects and a background image (Figure 10b). Converting the slide in Figure 10a to HTML resulted in one of the image objects being incorporated into the background image (Figure 10c). This new background image is potentially misleading as one would expect the representation shown in Figure 10d, where the users should be capable of selecting the image object. However, the user is presented with a misleading spatial representation (Figure 10e) as a result of this artifact. In the same way, the slide author's choice of layout could influence the effectiveness of the representation.


Computer Science Department

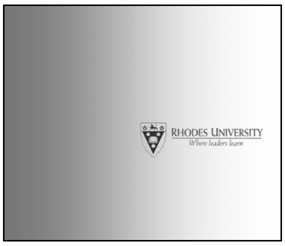
- CS Department
- http://www.cs.ru.ac.za/

- Rhodes University
- http://www.ru.ac.za/

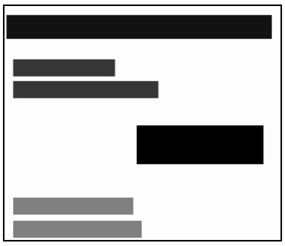
(a)



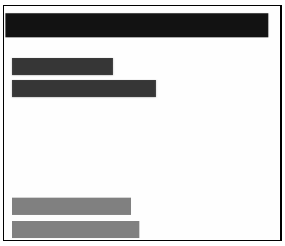
(b)



(c)



(d)



(e)

Figure 10. Limitations introduced by Microsoft PowerPoint. (a) The original slide. (b) The original background image. (c) The resulting background image after the conversion from PowerPoint to HTML where one image has been merged with the background image. (d) The expected spatial representation of the slide. (e) The actual, misleading representation.

5. Conclusions and Future Research

A proof-of-concept, content exposure and extraction system has been implemented for use with Microsoft PowerPoint slides shows that allows mobile device users to selectively annotate and download desired content. PowerPoint slideshows are automatically converted to HTML. The system then implements a spatial representation algorithm that has been shown to accurately represent individual slides irrespective of slide complexity. However, the algorithm is reliant on the author's chosen slide layout in order to produce an intuitive representation. A mobile web application allows users to navigate, annotate and save slideshow content. This application adapts to mobile device screen size, reducing the size of the spatial representation image in order to fit the application page into a single screen thus removing the need for scrolling. Testing has shown that the system can accommodate a maximum of approximately 118 simultaneous users transmitting approximately 35 requests per second. However, since the number of requests per second is expected to be much lower in a typical lecture environment, it is likely that the system can accommodate a larger number of users. Results from a user study indicated that users are able to learn how to use the system quickly, and consider the application to be a productive tool.

Future investigation will employ a more comprehensive user study in order to determine possible positive influences of the system on student learning. Third-party PowerPoint to HTML conversion tools exist and utilizing these tools may produce more accurate spatial representations. .NET Remoting provides the opportunity to further integrate the presentation system with the mobile web application by providing functionality for the applications to communicate. New releases of the Microsoft Office package, or other presentation software packages, may introduce further opportunities such as direct XML representation of slide shows.

References

Alam, H., and Rahman, F. (2003). Web Document Manipulation for Small Screen Devices: A Review. *Second International Workshop on Web Document Analysis (WDA2003),* Edinburgh, August 3.

Bartsch, R. A., and Cobern, K. M. (2003). Effectiveness of PowerPoint presentations in lectures. *Journal of Computers and Education*, 41, 77 – 86, Issue 1.

Buyukkokten, O., Garcia-Molina H. and Paepcke, A. (2001). Text Summarization of Web Pages on Handheld Devices. *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL),* Pittsburgh, PA, U.S.A., 2-7 June.

Chen, Y., Ma, W.Y., and Zhang, H.J. (2003). Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices. *Proceedings of WWW'03,* Budapest, Hungary, May 20-24.

Finn, M., and Vandenham, N. (2004). The Handheld Classroom: Educational Implications of Mobile Computing. *Australian Journal of Emerging Technologies and Society*, Vol. 2, No. 1.

Gupta, S., Kaiser, G., Neistadt, D. and Grimm, P. (2003). DOM-based Content Extraction of HTML Documents. *Proceedings of WWW'03,* Budapest, Hungary, May 20-24.

Gupta, S., Kaiser, G., Neistadt, D. and Grimm, P. (2004). Automating Content Extraction of HTML Documents. *World Wide Web Journal,* January.

Hadjisotiriou, C., Marshall, K., and Andrew, R. (2004). *ASP.NET Web Development with Macromedia Dreamweaver MX.* Apress Publishing, Berkeley, CA, USA. ISBN 1590593480.

Han, R., Perret, V., and Naghshineh, M. (2000). WebSplitter: A Unified XML Framework for Multi-Device Collaborative Web Browsing. *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW).*

Homer, A., Sussman, D., and Howard, R. (2005). *ASP.NET v. 2.0-The Beta Version, 2nd Edition.* Longman Publishing Co., Inc. Boston, MA, USA. ISBN: 0321257278.

ICESoft (2005). *ICEBrowser.* At http://www.icesoft.com/products/icebrowser.html

Kunze, M. and Rosner, D. (2001). An XML-based Approach for the Presentation and Exploitation of Extracted Information. *Proceedings of the First International Workshop on Web Document Analysis*, Seattle, WA, September 8.

Rogers, Y., Price, S., Fitzpatrick, G., Fleck, R., Harris, E., Smith, H., Randell, C., Muller, H., O'Malley, C., Stanton, D., Thompson, M and Weal, M. (2004). Ambient wood: designing new forms of digital augmentation for learning outdoors. *Proceedings of the 2004 conference on Interaction design and children: building a community table of contents*. Maryland, 3-10.

Roschelle, J. (2003). Keynote paper: Unlocking the learning value of wireless mobile devices. *Journal of Computer Assisted Learning*, 19, 260-272.

Senior Corps Tech Centre (2005). *Benefits of the Rich Text Format*. At http://www.seniortechcenter.org/reference_library/software/benefits_of_rich_text.php

Varshney, U. (2003). The status and future of 802.11-based WLANs. *IEEE Computer*, Vol. 36, No. 6, June.

Zurita, G and Nussbaum, M. (2004). A constructivist mobile learning environment supported by a wireless handheld network. *Journal of Computer Assisted Learning*, 20, 235-243.