Interactive Presentation Slide Adaptation Project

Interim Report: June 14, 2007

Vishvajit Singh
Supervised by Karl Martin and Prof. Kostas Plataniotis

Project Description

The slideshow presentation is widely used as a teaching tool, and banks of slideshow presentations are an invaluable resource for the independent learner. In order to further realize the potential of slideshow presentations for e-learning, we seek to create a content retrieval system that allows Microsoft PowerPoint slides to be viewed on a wide variety of mobile devices. Harnessing the ubiquity of mobile web browsers, we will create a web-based system that will serve PowerPoint slides in HTML/CSS form, with intelligent adaptations of the content so that it may easily be viewed on the small screen of the device. The user will then be able to interactively expand and probe the content as he or she sees fit.

Progress Report

I have achieved the following subgoals of this project:

- Research the Microsoft Office 2007 OpenXML format, PresentationML in particular.
- Make decision on full system architecture (detailed below).
- Make a C# preprocessing program that can read OpenXML packages, transform slide layout information to our own simplified XML representation, and extract images embedded in slides to a specified directory.
- Make an XSLT script that transforms our simplified XML representation of slides to a human-readable HTML web page, complete with CSS formatting.
- Write JavaScript code that is capable of detecting the size of the browser window, and resize the slides and slide content accordingly to fit the dimensions of the screen.

At present, the visual output of the system as viewed in the Firefox browser matches fairly accurately the basic elements of the input PowerPoint presentation. We are able to handle the placement of arbitrary text boxes (including slide titles and bullet points) and embedded images. The automatic resizing code works properly, and is capable of dynamically scaling text and images proportionally along with the entire slide.

The following tasks remain to be done:

- Process embedded vector graphics such as slide backgrounds into browser-readable SVG form.
- Completely finish the handling of slide layout information.
- Set up the Apache web server with PHP, and port the handling of the XSLT transform to PHP.
- Add interactivity to the browser-server interaction using asynchronous JavaScript (AJAX).
- Create a layout engine in JavaScript that is capable of intelligently rearranging slide elements to fit any size screen (rather than simply scaling all elements proportionally, as we are currently doing).
- Test application on actual or simulated mobile devices.
- Write documentation and finalize system, making it scalable, portable, and secure.

System Architecture

Server side *Apache web server running PHP 5*

The initial preparation of the system will require the administrator to run a C# preprocessing program on the input set of PowerPoint presentations. This need only be done once, on any Windows machine.

When the system is up and running, the PHP code on the server will be handling two different kinds of requests: slide requests and slide fragment requests. Slide requests will serve an entire slide of a presentation to the client, while slide fragment requests will respond to AJAX queries from the client with HTML representing an inner fragment of a slide. This is a central component of the interactive part of the system.

The PHP request handling code will be using XSL transforms to generate HTML/CSS output from the preprocessed representation of the presentations.

Client side *Web browser*

Upon accessing the system's website, the client will receive the first slide of the presentation being requested. Navigation arrows will allow him to proceed through the set of slides.

The representation of the individual slides will be adapted on-the-fly by the server to fit the size of the client's browser – that is, some elements of the slide will be 'minimized' into hyperlinks, which can then be clicked to view the full content. Client-side JavaScript code will carry out the fetching of this full content using AJAX.

Client-side JavaScript code will also handle the areas of slide adaptation that require real-time response to browser window resizing, such as the scaling and intelligent placement of slide elements.

Summary

The offline phase of the project is close to completion. We have two applications, one which processes PowerPoint presentations into an easy-to-parse intermediate representation, and one which takes this representation and produces human-readable output in a web browser. Now we have to move towards the online phase, and get this system working in the form of a server-side application. After the basic system is fully put together, we can start to focus on advanced features such as a JavaScript layout engine for intelligently restructuring slides to fit the small screens of cellphones and PDAs.