Using Microsoft PowerPoint Presentations To Create HTML5 Based E-learning Courses

Harsh Alkutkar, Ajinkya Abhyankar, Rushikesh Gawali, Saurabh Gandhele, Kailash Tambe

Abstract— This paper introduces how to use Microsoft PowerPoint presentation files to generate HTML5 based interactive e-learning courses packaged in SCORM format.

Index Terms— E-Learning, HTML5, Microsoft Powerpoint, SCORM

———————————— ◆ ————————————

1. INTRODUCTION TO E-LEARNING

The rapid development of Internet has allowed more and more people to obtain useful information via the Internet [1]. E-learning according to Markus(2008) can be defined as a learning process created by interaction with digitally delivered content, network-based services and tutoring support E -learning is any technologically mediated learning using computers whether from a distance or education or training to ICT-based personalized, flexible, individual, self-organized, collaborative learning based on a community of learners, teachers, facilitators, experts.[2] E-learning refers to the use of Internet technologies to deliver a broad array of solutions that enhance knowledge and performance. E-learning can be used by educators to improve the efficiency and effectiveness of educational interventions in the face of the social, scientific, and pedagogical challenges.

E-Learning modules typically consist of a welcome screen, section objectives, instructions, content screens, quizzes or challenges, assessment screens, summary of content and additional resources. [3]. These modules facilitate interactivity with technologies such as Flash or standalone exe's. These are packaged into a SCORM standard. SCORM stands for Sharable Content Object Reference Model.[4] SCORM is a suite of technical standards developed by the Advanced Distributed Learning (ADL) initiative to develop common specifications and standards for technology-based learning deployed over the internet. These standards enable web-based learning and content management systems to find, import, share, reuse, and export learning content in a consistent manner. In addition, it allows user tracking and reports to be generated based on learning objectives. Essentially, SCORM standardized the method of communication between eLearning courses and SCORM conformant learning and content management systems.

The advent of smartphone systems such ass iPhone, Android and Blackberry, which blend corporate and personal functions have

now allowed e-learning courses that were traditionally delivered online from personal computers to be delivered straight to mobile and tablet devices, allowing a greater degree of flexibility to the user , allowing one to take or continue the courses that were originally on his/her computer anywhere with a working internet connection from his/her mobile device or tablet.[5]

To create such content, there are several authoring tools available in the market. However we propose our own method to take a Microsoft Powerpoint 2007 Presentation File (.pptx) as Input, consisting of certain "marks" that we can interpret, and convert them into equivalent HTML5 based interactive courses, without the need for a costly third party tool.

2. OPEN XML FORMATS AND STRUCTURES OF POWERPOINT FILES

The 2007 Microsoft Office system adopts an XML-based file format for Microsoft Office PowerPoint 2007. The new file format, called Office Open XML Formats, addresses workplace challenges that included easily moving data between disparate applications, and allowing users to glean business insight from that data.[6] Many elements go into creating a Microsoft Office document. Some of these are commonly shared across all the Office applications, for example, document properties, style sheets, charts, hyperlinks, diagrams, and drawings. Other elements are specific to each application, like worksheets in Excel, slides in PowerPoint, or headers and footers in Word. By wrapping the individual parts of a 2007 Microsoft Office system file in a ZIP container, a document remains as a single file instance.

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards.[8]

Let us say we create a simple PowerPoint slide with some text and images as given below.

Fig.1 – The sample slide with the element numbers labeled in red. 1) Is a shape with a gradient. 2) Company logo. 3) Title Text 4) Content Text 5) Content Image 6) Next Shape with link

We save the presentation in the 2007 pptx extension on our hard drive. We then rename the .pptx to a .zip extension and extract the content to the hard drive. After unzipping we get three folders - *rels, docProps and ppt, and one file* [Content_Types].xml
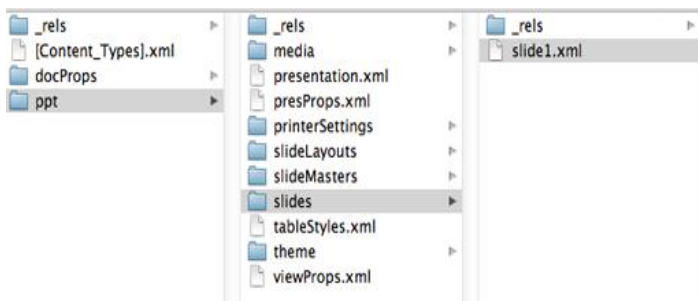

Fig-2 Content Structure of a .pptx file unzipped

If we open the *ppt* folder and browse to the *slides* folder we get to see all the contents of a slide in XML. In this case it was just one slide. A closer examination of the slide1.xml file reveals an organized structure of tags where the content is made accessible as shown in figure 3.

```
<p:txBody>
    <a:bodyPr>
        <a:spAutoFit/>
    </a:bodyPr>
    <a:lstStyle/><a:p><a:r><a:rPr lang="en-US" dirty="0"
        smtClean="0"/>
        <a:t>Selling shares in your company is one way of
            raising long-term finance for your business.
            This is also known as equity finance. The
            advantage of equity finance is that you don't
            always have to repay the finance or pay
            interest, as you would with an overdraft or bank
            loan.</a:t>
    </a:r>
    </a:p>
    <a:p>
        <a:endParaRPr lang="en-US" dirty="0"/>
</a:p></p:txBody></p:sp><p:sp><p:nvSpPr><p:cNvPr id="9"
    name="TextBox 8"/><p:cNvSpPr txBox="1"/><p:nvPr/></p:
    nvSpPr><p:spPr><a:xfrm><a:off x="1737736" y="434500"/
    ><a:ext cx="6449674" cy="830997"/></a:xfrm><a:prstGeom
    prst="rect"><a:avLst/></a:prstGeom><a:noFill/></p:spPr><
    p:txBody><a:bodyPr wrap="square"
    rtlCol="0"><a:spAutoFit/></a:bodyPr><a:lstStyle/
    ><a:p><a:r><a:rPr lang="en-US" sz="4800" dirty="0"
    smtClean="0"><a:solidFill><a:schemeClr val="bg1"/></a:
    solidFill></a:rPr>

    <a:t>Share Trading Course</a:t>
```

Fig -3 : A snapshot of the slide1.xml file containing the content wrapped around tags.

By looking for the <a:t> tags in our xml file we can easily capture content inside the slides and convert it into corresponding HTML5 templates. We can even include our own indicators such as "TITLE" or "BUTTON" inside the slide text itself , so the parser will interpret these tags and attach appropriate attributes to them in the HTML file.

3. SYSTEM IMPLEMENTATION

A. Overview of the working

We write a program which will accept the powerpoint file as input and convert it into a HTML5 file based on a predefined template that will be created to provide interactivity.
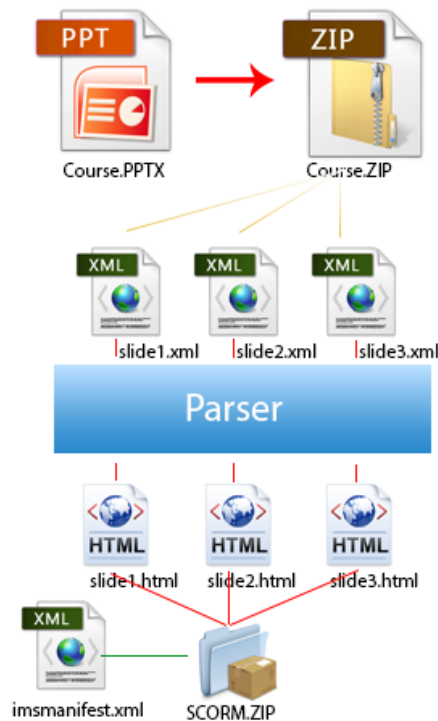

Fig -4 : A brief overview of the process

The first step will be to unzip the contents of the pptx file and open the slide.xml files for parsing. Once here, we can extract whatever tags we want from the XML file with the help of an XML API or simple file handling. In case we plan to handle only text we search for all the <a:t> tags inside the ppt file. We then map this input to another HTML5 file that has already been prepared and substitute the extracted values into the .html file. After finishing we link all the html files together and package them with a manifest file as required by the SCORM standard.

In case we want to parse images we can refer to the *media* folder and extract the actual images whenever they are encountered for example in the xml we encounter

<p:pic> […] <a:blip r:embed="rId3"/> […]

we consult the slide1.xml.rels file in the _rels folder that shows the relationship between the id and the media. The Relationship Id (rId2 in this case) simply provides a unique identifier for the referenced file.

><Relationship Id="rId3" Type="http://schemas.openxmlformats.org/officeDocument /2006/relationships/image" Target="../media/image1.png"/>

B. Algorithm

Input Preconditions: A Powerpoint file saved in Microsoft Open XML Format (.pptx). The Powerpoint presentation should contain slides with text and images, and optionally marked buttons for navigation of content such as "next" or "previous".

Process:

3.1. Program is invoked with three parameters- input file(.pptx) , html5 template (.html) and output file(.zip)

3.2. The input file (.pptx) is renamed to .zip and extracted to a temporary location say '/temp'

3.3. One file pointer is made to point to the first slide in ppt/slides/slide1.xml

3.4. Second file pointer is made to point to the corresponding rels file (slides/_rels/slide1.xml.rels)

3.5. For all slides do the following

    3.5.1. Find content within all <a:t> tags and put them into an array

    3.5.2. Within the same array for each element, search for recognizable tokens in the text such as "title" or "content"

    3.5.3. Remove the tokens from the original text and copy them into the html placeholders that hold text in the template.

    3.5.4. Search for <p:pic> tags and look for the relationship id such as "rId3"

    3.5.5. Using the second file pointer locate the path to the file with the given relationship id "rId3"

    3.5.6. Find this file in the ppt/media folder and attach it to the html5 template.

    3.5.7. Link the next / previous buttons to the previous or next slides with hyperlinks for the same.

3.5.8. Save the filled html template with a slideX.html name where X is the slide number

3.5.9. Read next element in array

3.6. Once all the html files have been generated and linked, create a top level manifest file called imsmanifest.xml

3.7. Fill in the necessary information in the manifest such as links to resources, initializations, launch conditions, time functions and so on.[7]

3.8. Zip the file and import it into the LMS or deploy the HTML content live on the website.

C. Outcome

The final outcome should be the working course with interactive buttons on an HTML5 based device as shown in the figure below.

Fig. 5 – Expected output of HTML5 based course running on an iPad

4. IMPROVEMENTS AND FUTURE WORK

Future improvements include parsing all types of Office documents such as Word and also parsing their original binary formats. We also are working on algorithms for greater functionality and interactivity such as drag and drop options, Voice Overs, Character animations and so on.

5. CONCLUSION

We can see from this paper that there is immense potential in harnessing Powerpoint based documents and converting them to easy HTML5 representations which can be packaged by the SCORM standard and deployed on LMS's.

6. AFFILIATION & AUTHOR INFORMATION

Harsh Alkutkar   [harshalkutkar@gmail.com]
Ajinkya Abhyankar   [aabhyankar9@gmail.com]
Saurabh Gandhele   [saurabhchel2117@gmail.com]
Rushikesh Gawali   [rushikeshkgawali@gmail.com]
Prof. Kailash Tambe   [kailashtambe@gmail.com]

All authors are affiliated with MIT College of Engineering, Pune, India.

REFERENCES

[1] Al-Zoube, M. (2009). *E-Learning on the Cloud. International Arab*

[2] Olojo Oludare Jethro - *E-Learning and Its Effects on Teaching and Learning in a Global Age* - (International Journal of Academic Research in Business and Social Sciences)

[3] Common Course Structures for e-learning - http://www.articulate.com/rapid-elearning/new-to-e-learning-here-are-some-simple-steps-to-help-you-get-started/

[4] Cyber Media Creations (CMC) - *SCORM 101* - http://www.cybermediacreations.com/elearning/white-papers/SCORM_101.pdf

[5] Andreas Holzinger - *Mobile Phones as a Challenge for m-Learning* , Proceedings of the 3rd Int'l Conf. on Pervasive Computing and Communications Workshops IEEE

[6] MSDN , *Introduction to Office Open XML Formats (MSDN)* - http://msdn.microsoft.com/en-us/library/office/aa338205(v=office.12).aspx

[7] Claroline Doc - *Make multiple pages an SCO* - http://doc.claroline.net/en/index.php/How_do_I_create_SCORM_content%3F

[8] XML wiki - http://en.wikipedia.org/wiki/XML