A transformation tool for adapting SCORM documents into Mobile environment

Chia-Feng Lin , Francis Lin and Shyan-Ming Yuan
{Department of Computer Science NCTU ; Department of accounting information Department of Information Management chihlee }
{ 1001 Ta Hsueh Road, Hsinchu, Taiwan 300, ROC ; 188, Sec. 5,Au-Chung Rd, Tainan, Taiwan}

Abstract: - Nowadays, the field of E-Learning becomes more and more popular. At the same time, M-Learning expand the ability of E-learning. Due to SCORM is the most widespread used technology for exchanging user's profile and learning resource under E-Learning platform, we will pay close attention to the appliance of SCORM on Mobile devices. In fact, the rule of presentation and navigation do not defined in SCORM specification, it dependents on the learning management server. In this paper, we major focus on how to adjusting SCORM content so that overcome the restriction while learning from mobile device. At the same time, we analysis the common cases when reading SCORM base document on Mobile device and design a transformation tool for adapting SCORM documents into Mobile environment.

.

Key-Words: - SCORM Mobile Transformation Learning Education XML

1   Introduction

Electronic learning (e-learning) becomes an important learning style for a long time. Recently, the issue about M-learning becomes more and more popular because of the improved on both computing power and memory storage for mobile device.

In our research, we will use a widely used E-learning sharable standard SCORM (Sharable Content Object Reference Model) which was developed by ADL [1] and supported by American government to executing on mobile device. We will discuss with the restriction while apply SCORM to mobile device and bring up our method to solve these problem. In addition, we collect many existing SCORM document developed by different organization and analysis what happen when they adjust to the size for pocket PC.

2   Problem description

2-1   The main problem
The architecture about "Packet SCORM" [4] and [5] implement not only the server side framework of LMS but also client side structure. Therefore, if we use their own authoring tool to editing a new course and execute on their LMS, these processes are sure to getting less problem. But if we need to export the course material to another LMS and the user still use

mobile device, we need to design a new SCORM document. It wasted time and resource. In this paper, we try to focus on how to utilizing the content side to reduce overhead for authoring a new document again. Next chapter will discuss about our system architecture and our consideration.

2-2 Analysis for SCORM Document

Table 1 Common SCORM document type analysis

Type	Action
Type1: Flash base type	Do nothing
Type2: Html base type	Do nothing
Type3: Chaotic type	Authoring this file
Type4: Unknown type	Do nothing

In order to transforming an existing SCORM format documents, we have already collect some common used SCORM document and take their snapshot. We try to summarizing the general type while adapting to small size. Table 1 list the result after we analysis.

2.2.1 Sample type description

Type1: Flash base type:

A Flash base presentation type usually integrated a Flash file into html document. Figure 1 is an example source code for Flash base document. In this document, html body only present a Flash file (named fofts2_component.swf). Figure2 shows original snapshot on PC, and Figure3 is a file after resize it's pixel to 300 * 200. In summary, Flash base document can not modify the presentation size. It needs sidelong scroll bar to browsing the page. Unless edit a new document which is suit for specific mobile device.

```
</HEAD>

<BODY bgcolor="#FFFFFF" background="../sharedassets/bg.jpg">

<center>
  <OBJECT classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
  codebase="http://download.macromedia.com/pub/shockwave/cabs/flash
  ID=allfunction WIDTH=615 HEIGHT=390>
    <PARAM NAME=movie VALUE="fofts2_components.swf">
    <PARAM NAME=quality VALUE=high>
    <PARAM NAME=menu VALUE=false>
    <PARAM NAME=bgcolor VALUE=#FFFFFF>
    <EMBED src="fofts2_components.swf" quality=high bgcolor=#FFF
    </EMBED>
  </OBJECT> <br>
  <font face="Verdana, Arial, Helvetica, sans-serif" size="1" colo
of Acronyms</a> | Text Only Alternate Pages for <a href="compone
  | <a href="#" onClick="MM_openBrWindow('../sharedassets/credits.
  This Course</font>
</center>

</BODY>
</HTML>
```

Fig1 A Flash base document source code.

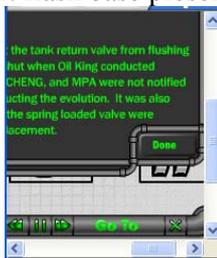Fig 2 Type 1: flash base presentation format

Fig 3 Type1:flash base format resized

Type2: Html base type

Html base presentation type stand for writing html code as simple as possible. In this way, we can calculate a page's width, and rewrite a page. For example, if a picture was putting in a html document and the width is not enough, we can take some process so that suit for reading. As show in Figure4

and Figure5, after resize the page's pixel, the document will adjust it self.

Fig4 Type2:html base presentation format

Fig 5 Type2: html base format resized

Type3: Chaotic type

Chaotic type means this kind of document can not be resizing. This is because the relative position in the document is fixed. May be they use many blank space to edit the document. After changing the pixel, the presentation becomes confused. The only way to overcome this drawback is write this course material again. Figure 6 and Figure 7 is the examples.
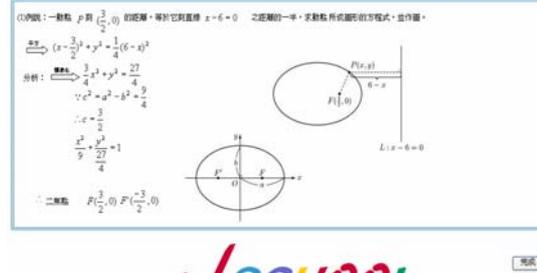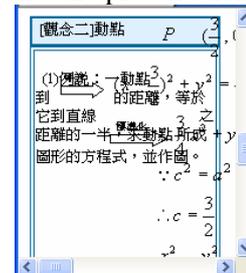

Fig6 Type3:Chaotiic presentation format

Fig7 Type3: Confused format after resized

Type4: Unknown type

Some SCORM document type was layout by specific mechanism, even we look into the source code, it hard to understanding how they write. They control the layout style not only using blank word. We do not research any more about this type because of the time.
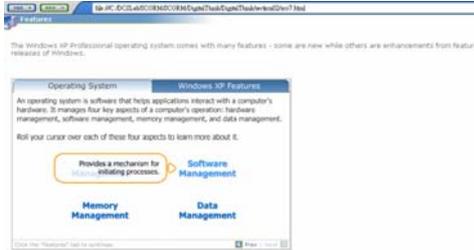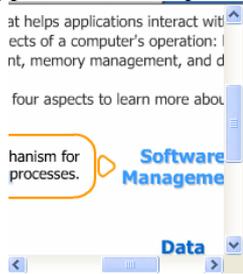

Fig Fig8 Type4:Unknown presentation format

Fi Fig9 Type4: Unknown format after resized

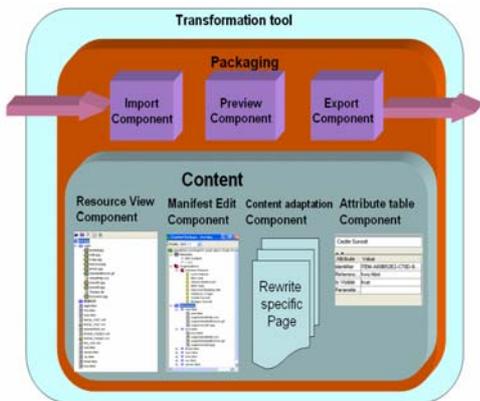## 3   Problem Solution

Fig 10 Toolkit architecture of our system.

The workflow of our transformation tool parsing a file with SCORM document format, do some authoring process and export the document. As depicted in Figure 10. In this chapter, each part of the architecture will be detail described
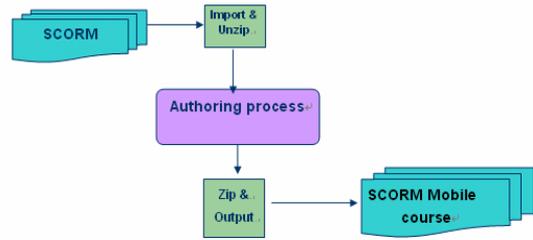
2-3 Toolkit composition
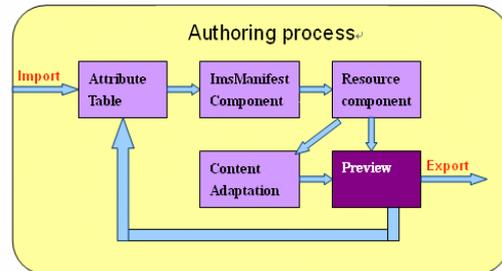

Fig 11 *Workflow of toolkit*

Fig 12 *Toolkit authoring process*

In face, a SCORM base document has two level ranking. First level is resource view which each file is unique. And then we transfer target file individually. We use resource component to handling it. The second level is manifest view which control by the file "imsmanifest.xml". This level identity the architecture of SCORM document, and we use Manifest Component to handling it.

Roughly,   our   system   does   follow   steps:
Step 1: Import an existing SCORM format document or create a new document.
Step 2: Configure system parameter.
Step 3: Edit the file "imsmanifest.xml", and add a new SCO head page (named top.htm) which includes original structure over original structure.
Step 4: Add hyperlink "previous page"、"next page" and "top page" to every.html(or .htm) format page.
Step 5: Apply "page rebuild" and "picture edit" for html format page
Step 6: Preview whole document from top page
Step 7: Export document or edit every html page from source code editor.

3-3-1 Attribute Table Component
In order to adapting for different kind of mobile device, attribute table configuration needs to access first. The most important thing here is pocket PC's pixel   setting.   With   appropriate   setting,   the transformation job con process well.

3.3.2 Manifest Edit Component

As show in Figure11, the left side of frame will provide   navigation   ability.   This   function   is convenient in general PCs but lack user friendly

while browse on mobile device. While browse on mobile device, this navigation portion waste space. In order to reducing the left navigation frame page, we parse the "imsmanifest.xml" file, change document structure as show in Figure12 and Figure3-6, it means we divided original page as show in Figure 3-4 into two pages.
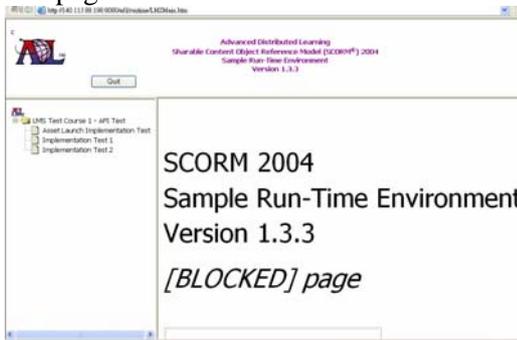

Fig 13 *The original page of SCORM SRTE*

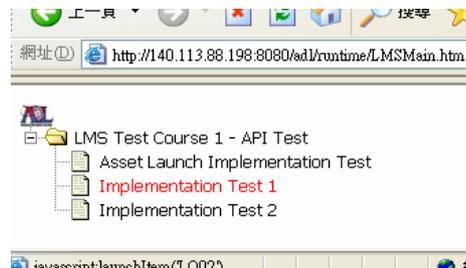Fig 14 *The snapshot after reduce left navigate page*

Fig 15 *The navigation of SCORM*

3-3-2 Resource View Component

Resource view component response for individual file level handling. In order to cooperate with Figure 13 and 14, we add three links in every .html format page in resource view. They are "navigation page", "previous page" and "next page" which enable user to browse between SCO and return to main page.
.

3-.3-3 Content Adaptation Component

There are many related transferring method about adapt to mobile device while browse web site. This is resource level modification. The Content Adaptation Component responsible for this portion. In fact, even the default browser, pocket IE, have support partially page adaptation.

3-3-4 Preview Component

We use preview mode to form a temporality html document and automatic open browser for examination with particular pixel size format. In preview mode, the JavaScript will also be supported so that it can simulate as if in mobile device.

3-4 Content Adaptation

SCORM specification does not define what kind of source file it should be, it only provides for the rule for communication and appear sequence. Therefore there are too many possibility that source will be. As the document presentation format analysis is chapter2.5, some document format can not be transformed well due to the original design. For example, a Flash base SCORM document is hard to decompose unless design on mobile device at beginning. In order to simplify our research, we actually deal with html postfix document. There are many research discuss with how to adapting document on mobile device. In our thesis, we will deal with picture first. Other further mechanism will not mention here.

3.4.1 Picture transform

While a picture's pixel size is large than pocket PC, it will cause difficult to browsing. Therefore, we provide the functionality to resizing this picture by editor.

3-5 Limitation for Mobile device

Due to the storage and computing power of mobile device becomes more and more powerful, some limitation before can be overcome now. After all, there are still some problem needs to be solve because of the difference between mobile device and PC. We will bring up some assumption to achieving the presentation of SCORM on mobile device. Include the connection problems, browser functionality support and storage limitation.

3.5.1 Connection problems

Due to the key point that we focus on handling content side first, we assume that the mobile device is always online to simplify the system complexity. In fact, the specification of SCORM supposed that the run time environment wad web base architecture, the LMS side need keep tracking client side's activity and refresh information. It is hard offline unless rewrite the LMS architecture. This is out of our objective. In this way, we suppose that server side and client side are always connected. We use pocket PC with wireless supported to keeping the connection.

3.5.2. JavaScript and Java Apple supportability

In general, the browsing functionality in mobile device seldom supported either JavaScript or Java Apple method due to the restriction of computing power and operating system support.

It's an unsolved problem in this paper so that we assume that the JavaScript will be design in future version of mobile device browser.
The first reason is that the operation of JavaScript immediate involve with LMS side. If we want to adjust the functionality of JavaScript, LMS side middleware must be added. This is out of our base key point to focus on content side.

The second reason is that after a series of testing, we find out the mobile device browse "Opera 8.5" could partially support this functionality. We suppose the newer version of the software will overcome the JavaScript restriction. The base requirement for SCORM is:
First, support JavaScript version 1.2.
Second, the browser support Java VM to load Java Applet.
Third, Java VM needs supporting the JavaScript calling Java Applet method.
Owing to reason one and reason two mention above, we use browser IE6.0 and execute on personal computer. But resize the display of browse to simulate mobile device display.

3.5.3. Storage limitation

We do not deal with temporary storage problem because we expect that the file size of single SCO is small. In general, SCORM document will be parsing in server side and then deliver single page as server-client mode. Unless the page it invoke external file such as movie or PowerPoint file, otherwise the document will not cause storage limitation.

4   Conclusion

There are growing amount of people start to enjoying distance education courses not only using their PCs but also using mobile devices such as Pocket PCs. Because Pocket PCs are easy to carry, these devices are considered as a proper platform for E-learning. In this thesis, we focus on the issue of transferring the current PC based Sharable Content Object Reference Model (SCORM) document to particular mobile device. Because we focus on document itself, there are related restriction such as connection needs to keeps alive and JavaScript functionality needs to be supported. We develop a transformation tool kit to re-write existing document. It saves developing time and makes the existing document more powerful.

The main purpose of crafting this toolkit is to reduce the efforts of developing applications for multiple mobile devices. For this reason, we discuss them in two criteria: rapidly development and write once, generate multiple application user-interfaces.

From the rapidly development aspect, it is important to reduce the efforts of learning how to use a new toolkit for developers. Therefore, the toolkit is provided with the simply steps as the interaction medium.

From the write once, generate multiple specific document aspect; it is completely feasible through our toolkit by using the transform technology. While a new mobile device appear, we only need to change the parameter setting and a new SCORM base document can be easily produced.

There are multiple improvements available for our system. First, since the version of SCORM 2004 supports sequencing control, most SCORM base document support this functionality. We need to do more discussion about sequencing control rules. Second, the JavaScript and Java Applet support need to solve so that our tool kit can really running on mobile device. Third, the media format support should be more clearly and efficiently, since the variety of SCORM document format.

5   References:

[1] http://www.adlnet.org   Advanced Distributed Learning (ADL) (2003)

[2] Universal Learning Format (ULF) Technical specification(2002)   Version   1.0, http://www.saba.com

[3] IMS Simple Sequencing Specification, IMS Global Learning Consortium, Inc. (2006), http://www.imsglobal.org.

[4] Nigel H. Lin, Timothy K. Shih, Hui-huang Hsu, Hsuan-Pu Chang, Han-Bin Chang and Wen Chieh Ko, "Pocket SCORM", Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW'04) 2004

[5] Reload editor http://www.reload.ac.uk

[6] SRTE http://www.adlnet.gov/scorm

[7] SCORM   Player http://www.reload.ac.uk/tools.html

[8] Carnegie Mellon University, Learning Systems Architecture Lab., SCORM Best Practices Guide for   Content   Developers, http://www.lsal.cmu.edu

[9] TK Shih, NH Liiit, HP Chang, KH Huang, Adaptive Pocket SCORM Reader Multimedia and   Expo, 2004. ICME '04. 2004 IEEE

International Conference on, 2004

[10] I. Waycott, and A. Kuhlska-Hulme (?003), "Students'experiences with PDAs for reading course materials" Personal and Ubiquitous Computing Volume 7, lssue 1 (May 2003), ISSN:1617-4909

[11] Kay Sommers, Jane Hcslcr, and Jim Bastick (ZOOI),"Little Guys Make a Big Splash PDA Projects at Virginia Commonwealth University", Proceedings of the 29th annual ACM SlGUCCS conference on User services (SIGUCCS' 01)

[12] Lehner, F.; Nosekabel, H., "The role of mobile devices in E-Learning first experiences with a wireless E-Learning environment", Wireless and Mobile Technologies in Education, 2002. Proceedings. IEEE International Workshop on , 29-30 Aug. 2002 Pages:103 – 106

[13] Staffan Bjork, Lars Erik Holmquist, Johan Redström, Ivan Bretan, Rolf Danielsson, Jussi Karlgren, Kristofer Franzén, "WEST: a Web browser for small terminals", November 1999, Proceedings of the 12th annual ACM symposium on User interface software and technology

[14] Jacob O. Wobbrock, Jodi Forlizzi, Scott E. Hudson, Brad A. Myers, "Novel input, output, and computation: WebThumb: interaction techniques for small-screen browsers", October 2002, Proceedings of the 15th annual ACM symposium on User interface software and technology