

# 國立交通大學

資訊科學與工程研究所

## 碩士論文

一個將 SCORM 文件轉換成適合於行動裝置閱讀格  
式的工具程式

A transformation tool for adapting SCORM documents into Mobile  
environment



研究生：林家鋒

指導教授：袁賢銘 教授

中華民國九十五年六月

一個將 SCORM 文件轉換成適合於行動裝置閱讀格  
式的工具程式

A transformation tool for adapting SCORM documents into Mobile  
environment

研究生：林家鋒

Student：Chia-Feng Lin

指導教授：袁賢銘

Advisor：Shyan-Ming Yuan

國立交通大學

資訊科學與工程研究所

碩士論文



Submitted to Department of Computer Science and Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

# 一個將 SCORM 文件轉換成適合於行動裝置閱讀格式的工具程式

研究生：林家鋒

指導教授：袁賢銘

國立交通大學資訊科學與工程研究所



目前透過電腦來完成學習的領域變得愈來愈熱門。在此同時，行動學習擴展了資訊教育所涵蓋的範圍。由於對於透過電腦交換使用者的學習歷程、學習資料的技術中，SCORM 是被使用最廣泛的。

因此我們將討論對於如何把 SCORM 應用在行動裝置中。事實上在 SCORM 的標準中，對於如何呈現及導覽並未做定義。在本論文中，我們主要集中在如何把 SCORM 的內容做適當的轉換，使得它能夠克服在行動裝置上瀏覽時所遇到的限制。在此同時，我們分析了閱讀 SCORM 格式的文件時常見的格式，在使用於行動裝置的時候遇到的狀況。並且設計了轉換工具來把 SCORM 文件轉換為適合行動裝置閱讀的格式。

# **A transformation tool for adapting SCORM documents into Mobile environment.**

**Student: Chia-Feng Lin**

**Advirsor: Shuan-Ming Yuan**

Institute of Computer Science and Engineering  
National Chiao Tung University

## **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 thesis, we major focus on how to adjusting SCORM content in order to 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 word: SCORM 、 Mobile 、 Transformation**

---

## Acknowledgement

---

能完成這篇論文，首先要感謝袁老師兩年來的指導與教誨，讓我了解到做研究的方法。並且在做研究的過程中獲益良多、更上一層樓。感謝林憲堂學長兩年來的各種訓練，以及不辭辛勞地為我解答各種問題，給我非常多的協助，讓我可以完成這篇論文。

感謝所有碩二的同學，兩年來一同成長一同學習。謝謝實驗室其他的同學與學弟妹，感謝你們的照顧，解答了許多課業研究上的問題。為實驗室帶來熱絡的研究風氣與實驗精神。



---

# Table of Contents

---

<b>Abstract.....</b>	<b>III</b>
<b>Acknowledgement.....</b>	<b>IV</b>
<b>Table of Contents .....</b>	<b>V</b>
<b>List of Figures.....</b>	<b>VII</b>
<b>Chapter 1 Introduction.....</b>	<b>1</b>
1.1 Preface.....	1
1.2 Motivation.....	1
1.3 Research Contribution .....	1
1.4 Outline of thesis .....	2
<b>Chapter 2 Background and Related Work.....</b>	<b>3</b>
2.1 Chapter introduction .....	3
2.2 Mobile Learning.....	3
2.3 SCORM Introduction.....	4
2.3.1 Content aggregation model (CAM) .....	5
2.3.2 SCORM1.2 & SCORM 2004 .....	12
2.3.3 Plugfest .....	12
2.4 Run Time Environment (RTE).....	13
2.4 Related research .....	15
2.4.1 Packet SCORM.....	15
2.4.2 CCU new architecture.....	17
2.4.3 Opera8.5 for mobile device.....	18

2.5 Document Presentation Format Analysis.....	19
2.6 Discussion and Conclusion.....	23
<b>Chapter 3 System Architecture.....</b>	<b>24</b>
3.1 Chapter Introduction.....	24
3.2 System overview.....	24
3.3 Toolkit composition.....	25
3.3.1 Attribute Table Component.....	26
3.3.2 Resource View Component.....	27
3.3.3 Content Adaptation Component.....	28
3.3.4 Preview Component.....	28
3.4 Content Adaptation.....	28
3.4.1 Picture transform.....	29
3.5 Limitation for Mobile device.....	29
3.5.1 Connection problems.....	29
3.5.2. JavaScript and Java Apple supportability.....	30
3.5.3. Storage limitation.....	31
<b>Chapter 4 Implementation.....</b>	<b>32</b>
4.1 Developing environment – “Reload Editor”.....	32
4.2 System view.....	32
4.2.1 Resource view component modification.....	33
4.3 Rte view.....	34
4.4 Discussion.....	35
<b>Chapter 5 Conclusion.....</b>	<b>37</b>
5.1 Conclusion.....	37
5.2 Future work.....	38
<b>Chapter 6 Reference.....</b>	<b>39</b>

---

## List of Figures

---

Fig 2-1	Example of Content Aggregation Model .....	5
Fig 2-2	Content Model – Asset .....	7
Fig 2-3	<i>Figure 2-3 Content Model – SCO</i> .....	7
Fig 2-4	<i>Content Model – Content Aggregation</i> .....	8
Fig 2-5	<i>Content packaging conceptual diagram</i> .....	10
Fig 2-6	Overview of content packaging in the file “imsmanifest.xml” .....	11
Fig 2-7	<i>The process of API adapter</i> .....	14
Fig 2-8	<i>Run-Time environment</i> .....	15
Fig 2-9	<i>The system architecture of Packet SCORM</i> .....	16
Fig 2-10	<i>The run time snapshot of Packet SCORM</i> .....	17
Fig 2-11	The result of use mobile device connecting to LMS server .....	18
Fig 2-12.	<i>SCORM-based e-learning Environment for Mobile device architecture</i> .....	18
Fig 2-13	A Flash base document source code. ....	20
Fig 2-14	<i>Type 1: flash base presentation format</i> .....	20
Fig 2-15	<i>Type1:flash base format resized</i> .....	20
Fig 2-16	<i>Type2:html base presentation format</i> .....	21
Fig 2-17	<i>Type2: html base format resized</i> .....	21
Fig 2-18	<i>Type3:Confused presentation format</i> .....	22
Fig 2-19	<i>Type3: Confused format after resized</i> .....	22
Fig 2-20	<i>Type4:Unknown presentation format</i> .....	22
Fig 2-21	<i>Fig2-21 Type4: Unknown format after resized</i> .....	23



Fig 3-1 Toolkit architecture of our system.....24

Fig 3-2 *Workflow of toolkit* .....25

Fig 3-3 *Toolkit authoring process* .....25

Fig 3-4 *The original page of SCORM SRTE*.....27

Fig 3-5 *The snapshot after reduce left navigate page*.....27

Fig 3-6 *The navigation of SCORM* .....27

Fig 4-1 *Snapshot of Reload Editor* .....32

Fig 4-2 *The plug-in snapshot of our toolkit* .....33

Fig 4-3 *The snapshot of source code editing function* .....33

Fig 4-4 *Example SCO for modification* .....34

Fig 4-5 *top.html view* .....35



---

## List of Tables

---

Table 1 The definition of SCORM API. ....	14
Table 2 Common SCORM document type analysis .....	19



---

# Chapter 1 Introduction

---

## 1.1 Preface

Electronic learning (e-learning) becomes an important learning style for a long time. Recently years, the issue about M-learning becomes more and more popular because of the improvement on both computing power and memory storage for mobile device. M-learning is a development from E-learning which for its part originates from E-learning. The rapid growth of information and communication technologies makes it possible to develop new forms of this education. Today's student knowledge of mobile devices makes the entrance of mobile learning possible.



## 1.2 Motivation

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.

## 1.3 Research Contribution

A toolkit is designed and implemented based on these objectives, there are many

problems encountered in the process. This paper provides an impermanent solution to solve those problems. For major contributions of this paper are listed below:

1. We debate with related research about applying SCORM document into Mobile device. And then we figure out their advantage and limitation.
2. Extend SCORM usability: In the past, the SCORM document is design for normal personal computer (PC) which had enough display space and specific multimedia function support, but in Mobile World it does not well supported. In this paper, we will take some method to overcome this situation.
3. Reduce developing time. Editors do not need to authoring a new document for mobile device version while using our transformation tool kit.
4. Simply to use: With the implementation of graph user interface platform, a new user will be quickly familiar with the transformation process.

## 1.4 Outline of thesis



This thesis is composed of five chapters. Chapter 1, this one, is introductory.

In Chapter 2, we introduce the background of SCORM and related works about applying SCORM mobile devices. In the meanwhile, pro and con of various design decisions is also debated here. In Chapter3, we confer the detail design of our System Architecture. In Chapter 4, we presented our transformation tool and showing up the process of how to use it. Finally, in Chapter 5, we bring some conclusion and brief discussion on future works.

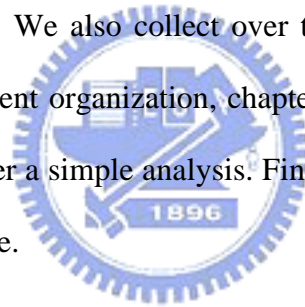
---

# Chapter 2 Background and Related Work

---

## 2.1 Chapter introduction

In this chapter, we introduced what was SCORM in our point of view. Chapter 2.2 introduced about Mobile Learning, which is more and more popular recently. In chapter 2.3, we brief introduced the SCORM specification whose is related to our work portion. In chapter 2.4, SCORM Run-time environment was presented here due to SCORM document is tightly interacted with RTE. In the last few years, several articles have been devoted to the study of applying SCORM to mobile device. In chapter 2.5 we discuss with related research. We also collect over twenty different existing SCORM document published by different organization, chapter 2.6 will present the result while adjust to small size layout after a simple analysis. Finally, in chapter 2.7 we give a brief discussion and conclusion here.



## 2.2 Mobile Learning

The advancement for mobile device and wireless has growing in an amazing rate. Almost everyone have a mobile phone or PDA today. The ability of computing power and software supporting for mobile device are more and more close to personal computer. In the mean time, 3G and Wireless LAN become popular everywhere. Mobile Learning is next generation of E-learning, not only substitute for Mobile Learning. However, learning anywhere and anytime is not all of Mobile learning. While transfer from E-learning to Mobile learning, we may make some mistake. For example, if we transfer a content presented in PC being to mobile device without any adaptation, there may be getting into some trouble. Similarly, if we use the same learning material but

different input equipment (e.g. no keyboard for input), what should we change?

## 2.3 SCORM Introduction

SCORM is a collection of standards and specifications adapted from multiple sources to provide a comprehensive suite of e-learning capabilities that enable interoperability, accessibility and reusability of Web-based learning content. There are many groups researching about sharing mechanism for teaching material, and has published many teaching material standard. For example, Sharable Content Object Reference Model(SCORM) has published by ADL( Advanced Distributed Learning), Universal Learning Format(ULF) [] has published by Saba, Aviation Industry CBT committee(AICC)'s Computer Managed Instruction(CMI) data model[], IMS( Instruction Management System) Global Learning Consortium's teaching material standard[] and Learning Object Metadata(LOM) publish by IEEE Learning Technologies Standardization Committee(LTSC)... etc. Along the knowledgeable teaching material's expansion, SCORM becomes the most complete and the most cited standard. ADL hope to building a sharable, reusable and interoperable teaching material developing mechanism to reduce course developing time and cost. Besides, all on-line teaching material could play on different learning platform and learning device easily. SCORM emphasize that base on WBT (Web-based Training) and cite current specification from AICC, IMS, IEEE/LTSC... etc, integration by XML and Java. These standards suit for the follow demands.

- (1) Reusable: learning content can modify and reuse easily in different tool.
- (2) Accessible: learning object can be searching in different system platform.
- (3) Durable: when system software changed, no necessary to modify program and

teaching material.

(4) Interoperable: teaching material can used on different platform.

(5) Adaptable: adjust learning content according to user's situation.

(6) Affordable: make up teaching material in a low cost way.

In order to a standardizing course authoring tool, SCORM defines content aggregation model (CAM) and run-time environment (RTE). They already has mature specifications, we will give a brief introduction as follow

### 2.3.1 Content aggregation model (CAM)

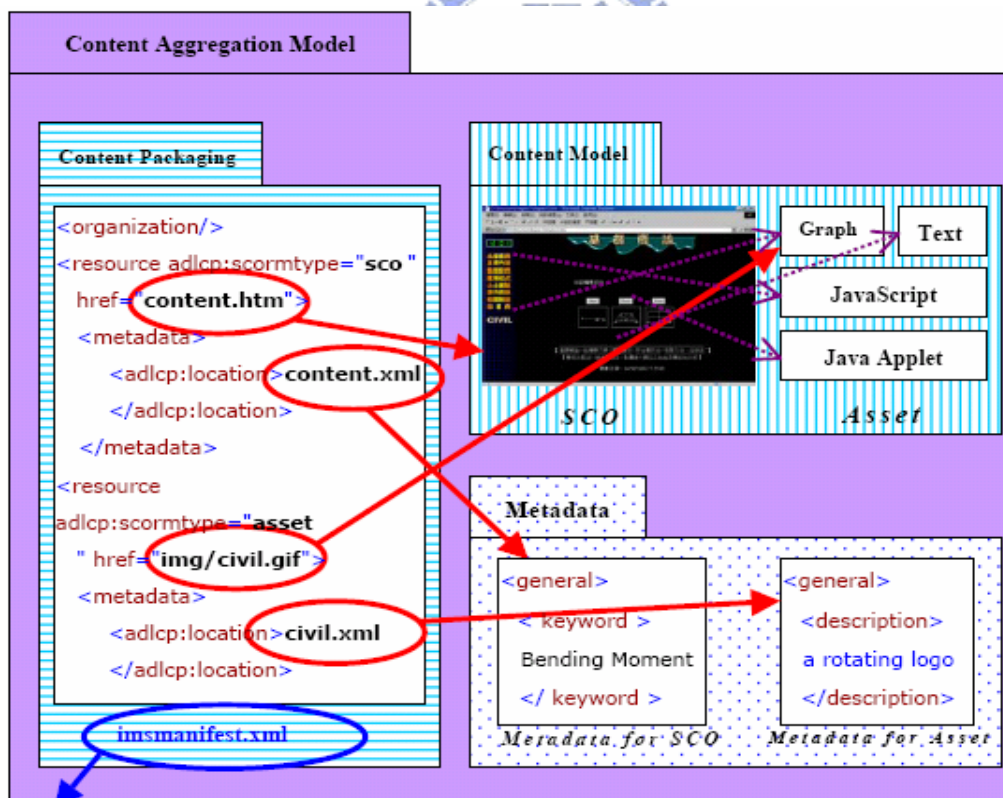


Fig 2-0-1 Example of Content Aggregation Model

Figure 2-1 is an example of CAM. Content aggregation model provides how to describe

and define a learning content, and describe how to combine different learning subject with a course. In this way, the course becomes a shareable learning resource, and can be applying into different learning management system and learning repository []. CAM includes three subjects: (1) Content Model: the definition of SCORM learning object content model.

(2) Meta-data: the information of learning elements.

(3) Content Packaging: it defines how to packaging and combine several learning element as a course structure format (CSF)

The detail of CAM can refer to the specific “SCORM Content Aggregation Model” publish by ADL.

We will give a brief introduction as below:

Advanced Distributed Learning (ADL) initiative, “SCORM Specifications – SCORM Version 1.3 Application Profile Working Draft Version 1.0”, March 26, 2003



#### 2.3.1.1 Content model

Content model includes three kinds of learning objects, they are Asset, SCO and Content Aggregation.

##### A. Asset

Asset are the basic element in learning objects, they includes Movie, Text, Picture, Sound, html files which can be transfer as web client format.



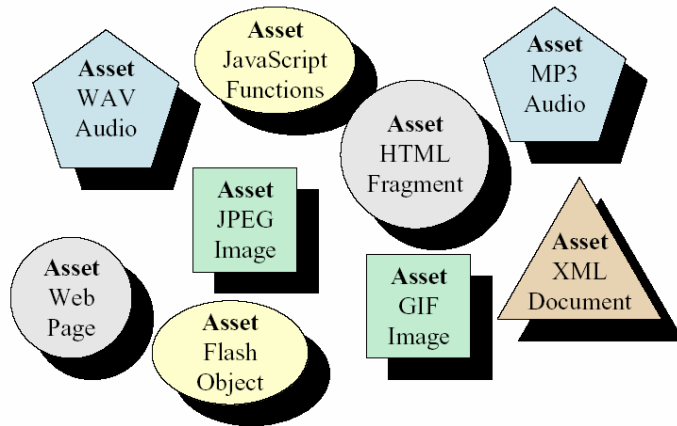


Fig 2-0-2 Content Model – Asset

### B. Sharable Content Object (SCO)

SCO are the collection of one or several assets, in addition, they includes the ability to communicate with LMS. SCO are the primitive learning object which can be tracking by run-time environment, and the smallest subject can be learned from LMS. A concept of SCO is showing as figure?.

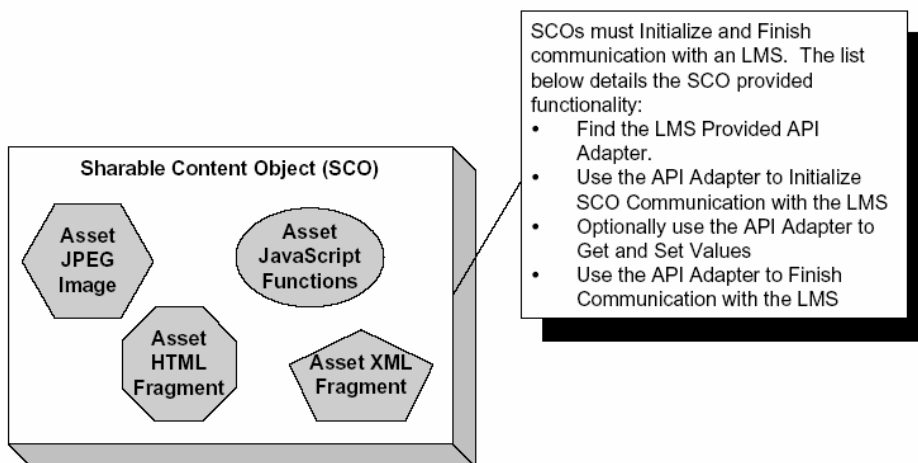


Fig 2-0-3 Figure 2-3 Content Model – SCO

SCO could operate API via API Adapter (the element who implement API), at least include the API LMSInitialize (“”) and LMSFinish (“”), which is the ability to opening and closing SCO. We will illustrate the method to communication with LMS in chapter

## 2.4

### C. Content Aggregation

As the picture ? Content Aggregation are the course structure of content, they includes several SCO or Asset, and turn into a meaningful learning unit. (e.g. a chapter in a course) They provide the LMS for designing the course material while combine with the Content Aggregation Meta-data whose describe the sequence of SCO or Asset in these learning elements, and becomes a complete course. In brief, Content Aggregation is also called Content Structure.

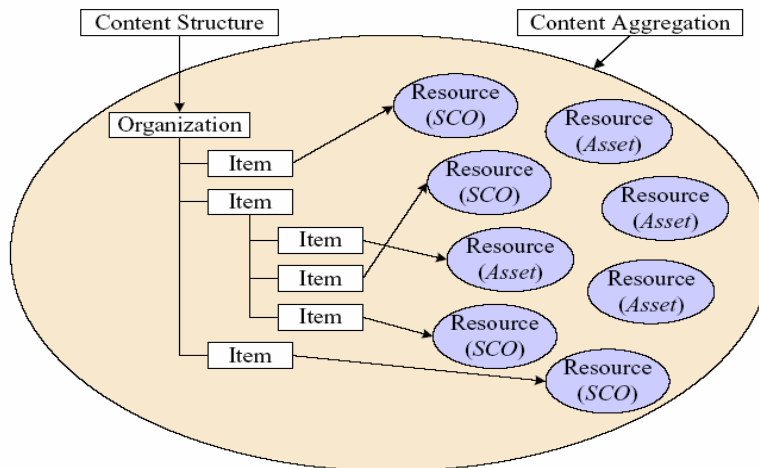


Fig 2-0-4 Content Model – Content Aggregation

#### 2.3.1.2 Meta-data

The SCORM Meta-data Profiles represents a mapping and recommended usage of the IEEE Learning Technology Standards Committee (LTSC) Learning Object Metadata (LOM) elements for each of the SCORM Content Model Components. In general, guidance is provided for meta-data to be applied to Assets, SCOs, Activities, Content Organizations and Content Aggregations to describe them in a consistent fashion such that they can be identified, categorized, searched for and discovered within and across

systems to further facilitate sharing and reuse.

The mechanism for binding the Content Model components that were discussed earlier to the matching Meta-data application profile is the Content Package as described in SCORM. There are currently five places meta-data can be applied within a content package:

- **Manifest:** Meta-data at the manifest level describes the content package as a whole. Meta-data placed at the manifest level is SCORM *Content Aggregation Meta-data*.

- **Organization:** Meta-data at the organization level describes the Content Organization as a whole. This may be a course, unit, lesson or any other organized instructional unit. Meta-data placed at the organization level is SCORM *Content Organization Meta-data*.

- **Item:** Meta-data at the item level describes a nested hierarchy of Activities in a context sensitive manner. When associated with an item, the SCORM *Activity Meta-data* definition must be used.

- **Resource:** Meta-data at the resource level describes a SCO resource or an Asset resource in a context insensitive manner. This meta-data is bound by either the SCORM *SCO Meta-data* or *Asset Meta-data* definitions (Determined by the type of resource - `adlcp:scormType` attribute).

- **File:** Meta-data at the file level describes an Asset in a context insensitive manner. This meta-data is bound by the SCORM *Asset Meta-data* definition.

### 2.3.1.2 Content packaging

Once learning content is designed and built, there is a need to make the content available to learners, authoring tools, repositories or Learning Management Systems (LMSs). The IMS Content Packaging Specification was designed to provide a standard

way to structure and exchange learning content.

The purpose of the Content Package is to provide a standardized way to exchange learning content between different systems or tools. The Content Package also provides a place for describing the structure (or organization) and the intended behavior of a collection of learning content.

Figure 2-5 is a conceptual diagram that illustrates the components of a Content Package.

A Content Package contains two major components:

- A special XML document describing the content structure and associated resources of the package called the manifest file (`imsmanifest.xml`). A manifest is required to be present at the root of the content package.
- The physical files making up the content package.

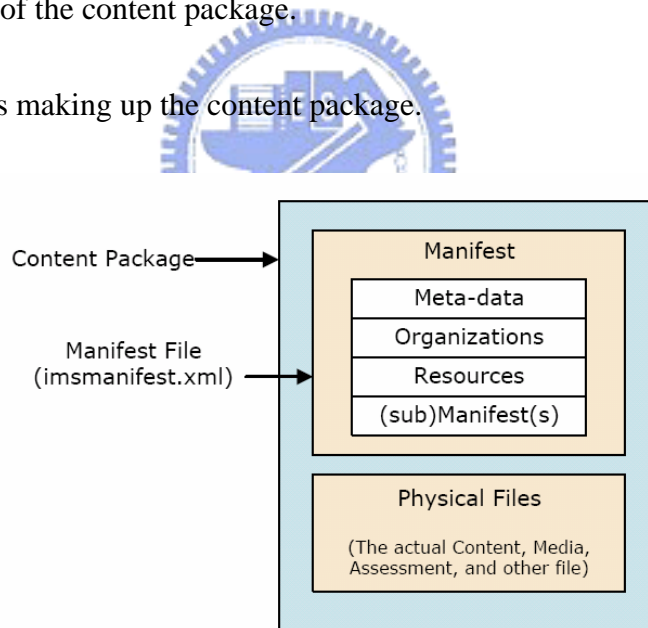


Fig 2-0-5 Content packaging conceptual diagram

Figure 2-5 is also called Package Interchange File (PIF). PIF is a binding of the content package components in the form of a compressed archive file. The PIF contains the `imsmanifest.xml`, all control files and the resources referenced in the content package (those that are local to the PIF, i.e., contained in the content package). SCORM recommends that content packages be created as PIFs. The PIF provides a concise Web

delivery format that can be used to transport content packages between systems.

### 2.3.1.3 imsmanifest.xml

As show in figure 2-5, the tag `<organization>` defines the relationship between course structure and each learning element hierarchy. Tag `<item>` is the learning element whose attribute “`identifierref`” map to `<resource>` (e.g “ I ” in figure 2-6). Tag `<resource>` point out learning element’s physical files location. Tag `<adlcp:location>something</adlcp:location>` (e.g “ II ”in figure 2-6) would link to SCO’s or Asset’s meta-data XML file. The detail argument for `imsmanifest.xml` can refer to the document SCORM CAM chapter 3.4[]

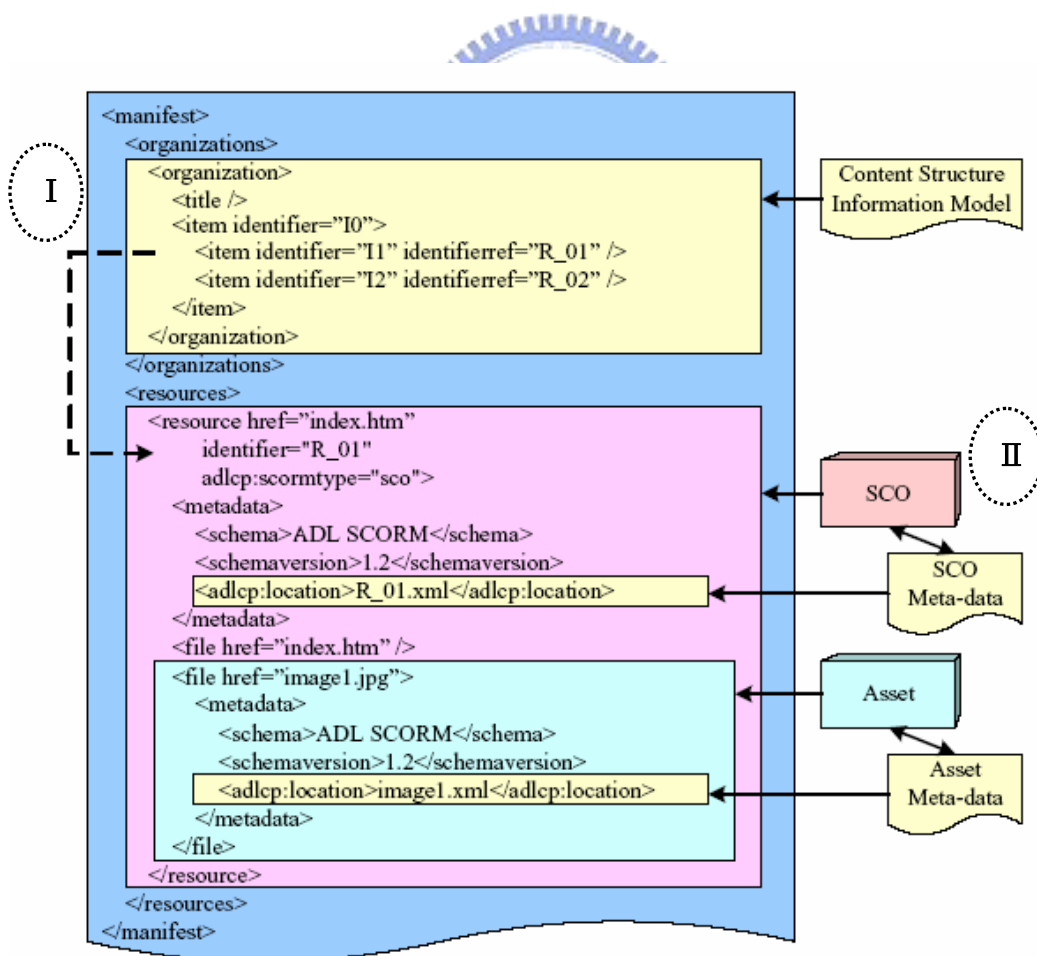


Fig 2-0-6 Overview of content packaging in the file “imsmanifest.xml”

#### 2.3.1.4 Sequencing

For different use, it will be more powerful if providing different learning activity sequencing. IMS bring up SSS (simple sequence specification) to enhance the power of SCORM. It includes a series of different kind sequencing rule, e.g. “No Sequencing rules”, “Linear”, “Constrained”, “Knowledge paced” and “Remediation”. All of them will be described by the file “imsmanifest.xml”. In our thesis, we do not spend time to do more discussion about sequencing.

#### 2.3.2 SCORM1.2 & SCORM 2004

In fact, the SCORM specification is still upgrading now. The newest version of SCORM is SCORM2004 while the most popular version before is SCORM version 1.2. The mainly different between SCORM2004 and SCORM1.2 is Content Aggregation Model and Run-time Environment. In Content Aggregation Model, they add the sequencing rule so that is becomes more complex. As to the Run-time environment, IEEE team modified some API and model. If somebody want to upgrade a course from SCORM1.2 to SCORM2004 without adding sequencing rule, the most important point is the Run-time environment portion. This is because the learning record will get into some trouble. Actually, this difference can be solved by program.

#### 2.3.3 Plugfest

Because ADL only publish the specification of SCORM, every organization can implement their LMS and SCORM document. In order to testing whether a SCORM teaching material produced by organization A can suit for LMS design by organization or opposed side. ADL hold a platform named “Plugfest” annually. In this place is the

best time to verify their product.

## 2.4 Run Time Environment (RTE)

Except for CAM, SCORM also define the run time environment. It includes three parts, (1) Launch mechanism: how the learning management system startup up learning content (2) Application Program Interface (API): the communication model between learning management system and learning elements (3) Data Model: the data format to communication between learning management system and SCO. We will give a brief introduction as follow.

### (1) Launch mechanism

Every learning management system follow the model “Launch” startup the learning resource which is suit for SCORM data type through API.

### (2) Application Program Interface (API)

API could show LMS the SCO’s state. Learning elements could also obtain the parameter for showing material resource through API. For example, require student’s information from LMS, and according to student’s grade and ability providing for different level learning material. SCORM define 8 kinds of different API as show in Figure 2-7

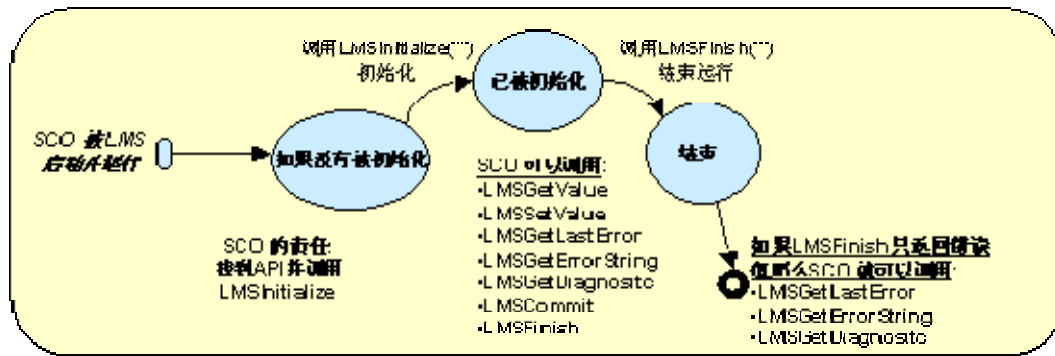


Fig 2-0-7 The process of API adapter

Table 1 The definition of SCORM API

	API	Function	Needed
<b>Execution state</b>	LMSInitialize	Learning resource open the communication with LMS	V
	LMSFinish	Learning resource close the communication with LMS	V
<b>Data transmission</b>	LMSGetValue	Request some parameter in Data Model from LMS	
	LMSSetValue	Setting some parameter in Data Model from LMS	
	LMSCommit	After using LMSSetValue, ask LMS store up the value in cache	
<b>State management</b>	LMSGetLastError	Getting error message from last API	
	LMSGetErrorString	Getting the error message description	
	LMSDiagnostic	Getting LMS's internal error message description	

According to the course structure configure in content package, LMS present course material in user's browser, and follow by sequence( Refer to imsmanifest.xml in tag <organization> configuration ) to launch Asset or SCO.

For the type of Asset, LMS will present directly through HTTP protocol. The communication between SCO and LMS will be control by API mechanism.

SCO communicate with LMS through calling API, such as calling LMSInitialize API when the communication starts and calling LMSFinish API when communication is ending. They also call LMSGetValue API and LMSSetValue API to exchange some data as data model format between LMS and



SCO. The SCORM specification also stipulate that LMS must provide “API Adapter” to implement these API function. While a SCO were browsed, it would hierarchical searching the API Adapter location recursively for taking API from the web page’s location. In this way, course editor do not need to take care the communication problem when author teaching material. They only needs to add the user interface such as JavaScript files provide by LMS to applying API. The architecture is show as figure 2-8.

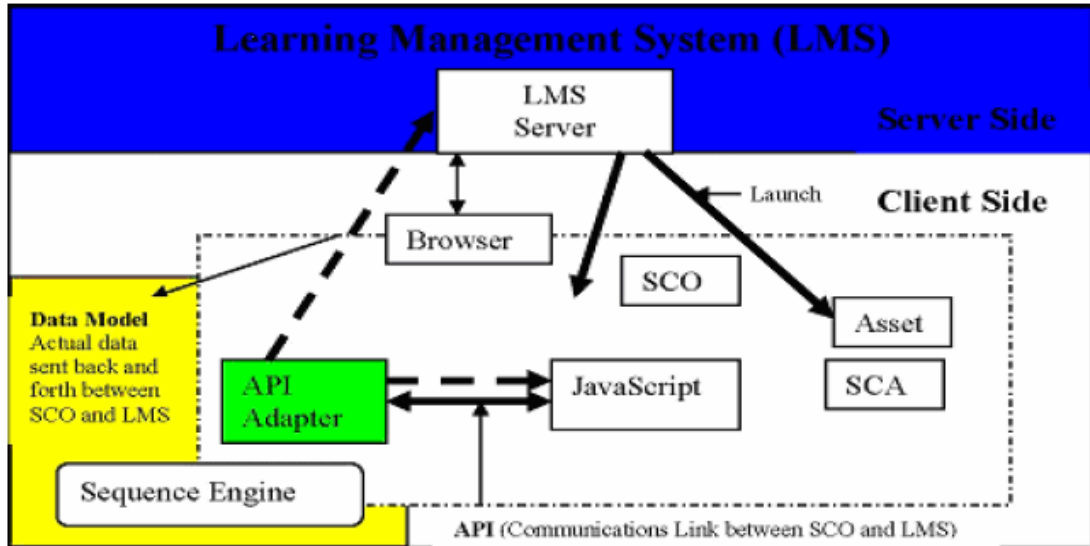


Fig 2-0-8 Run-Time environment

### (3) Data model

Data model is a common data format for transmitting data between LMS and learning resources. With this common format, LMS could control learning resource’s state or student’s learning portfolio. SCORM integrate AICC CMI Data Model standard specification, and define eight kinds of data type. In our thesis, data model do not take into consideration.

## 2.4 Related research

### 2.4.1 Packet SCORM

In year 2004, Lin published the Packet SCORM [4] prototype. They build their own

SCORM LMS Server cooperating with specific packet SCORM RTE in mobile device side and show in Figure 2-9 and Figure 2-10. The added more assist function to keep the data while the server is offline and lack of storage. But they only published a prototype especially focus on client side. The evaluation and detail about this system is not yet published so that we only have an elementary draft. We can not find out how they solve the problem about JavaScript with Java Applet support. Without JavaScript support, this system will as like a set of static html files. Besides, they do not discuss about the accept rate and problem while exporting their course to other LMS and importing course from other LMS.

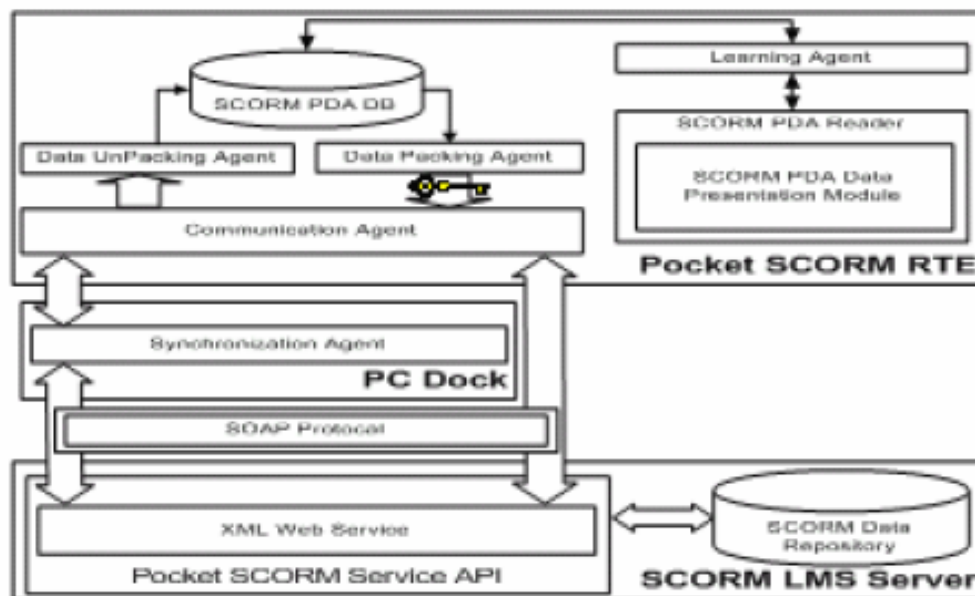


Fig 2-0-9 The system architecture of Packet SCORM



Fig 2-0-10 The run time snapshot of Packet SCORM

In [7] discuss about the reflow method for Packet SCORM. Reflow mechanism is an important function for Packet SCORM which increase the readability of the mobile device. In our paper will use their method to reflow SCORM document.

## 2.4.2 CCU new architecture

In year 2002, Professor Hsu published a SCORM-based e-learning Environment for Mobile device [5]. They discuss about due the lack of JavaScript supporting, if somebody use a mobile device connecting to LMS server, the result will as like the Figure 2-11.

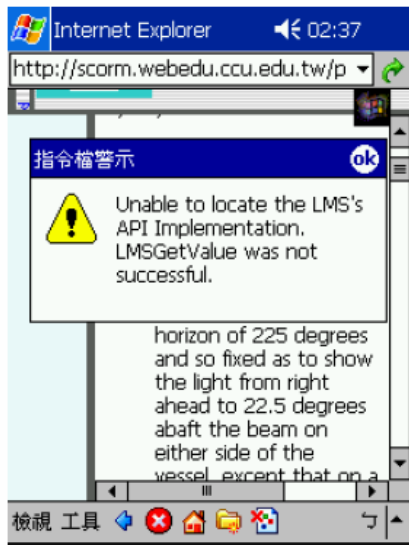


Fig 2-0-11 The result of use mobile device connecting to LMS server

In order to solve this problem, they bring up a new architecture as show in Figure 2-12.

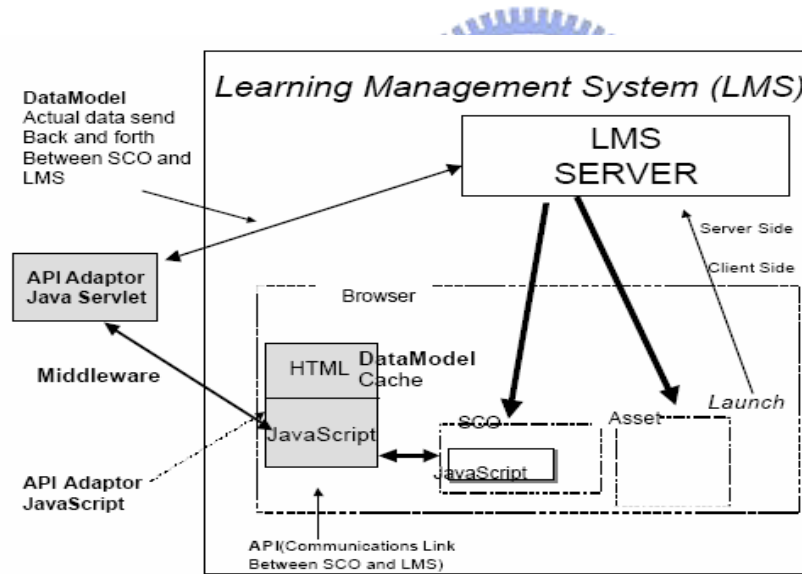


Fig 2-0-12. SCORM-based e-learning Environment for Mobile device architecture

### 2.4.3 Opera8.5 for mobile device

We try to finding out a real mobile device that supporting the JavaScript ability. From now on, the most suitable mobile device browser we find is “opera8.5 for mobile device version”. It almost supports every function, but still gets into some trouble that can not present the document on client side well.

## 2.5 Document Presentation Format Analysis

In order to transform an existing SCORM format documents, we have already collect some common used SCORM document with taking their snapshot. And that try to summarizing they general type while adapting to small size. Table 2-1 list the result after we analysis.

Table 2 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

### 2.5.1 Description for all of these sample type

- Type1: Flash base type:

A Flash base presentation type usually integrated a Flash file into html document. Figure 2-13 is an example source code for Flash base document. In this document, html body only present a Flash file (named fofts2\_component.swf ). Figure 2-14 shows original snapshot on PC, and Figure2-15 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>

```

Fig 2-0-13 A Flash base document source code.

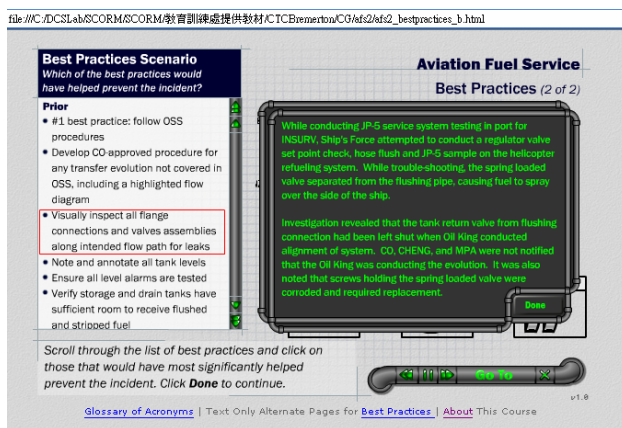


Fig 2-0-14 Type 1: flash base presentation format

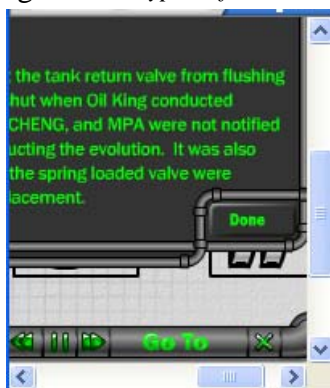


Fig 2-0-15 Type 1: flash base format resized

- 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 Figure2-16 and Figure2-17, after resize the page's pixel, the document will adjust it self.



Fig 2-0-16 Type2:html base presentation format



Fig 2-0-17 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. Maybe 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 18 and Figure 19 is the examples.

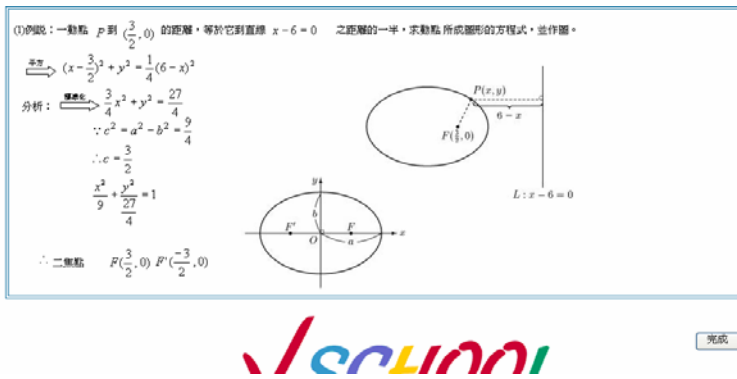


Fig 2-0-18 Type3: Confused presentation format

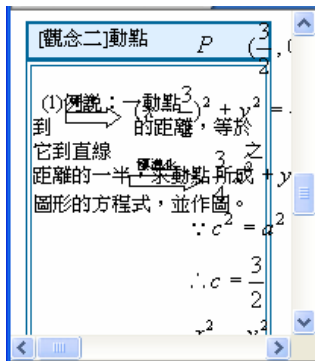


Fig 2-0-19 Type3: Confused format after resized

- 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.

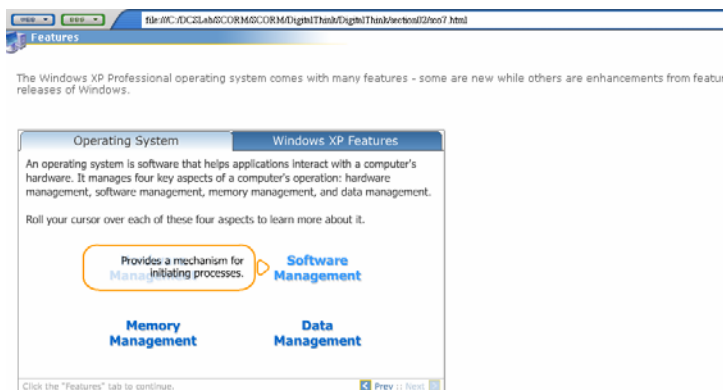


Fig 2-0-20 Type4: Unknown presentation format



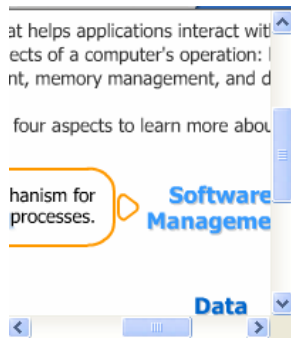


Fig 2-0-21 Fig2-21 Type4: Unknown format after resized

## 2.6 Discussion and Conclusion

The system 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 method of thinking.

---

# Chapter 3 System Architecture

---

## 3.1 Chapter Introduction

In this chapter, we will introduce the framework of our system. Ch 3.2 discuss with our system architecture. Ch3.3 discuss with the toolkit composition of every component. Include the flow chart of transformation process. Ch3.4 introduced the content adaptation component.

## 3.2 System overview

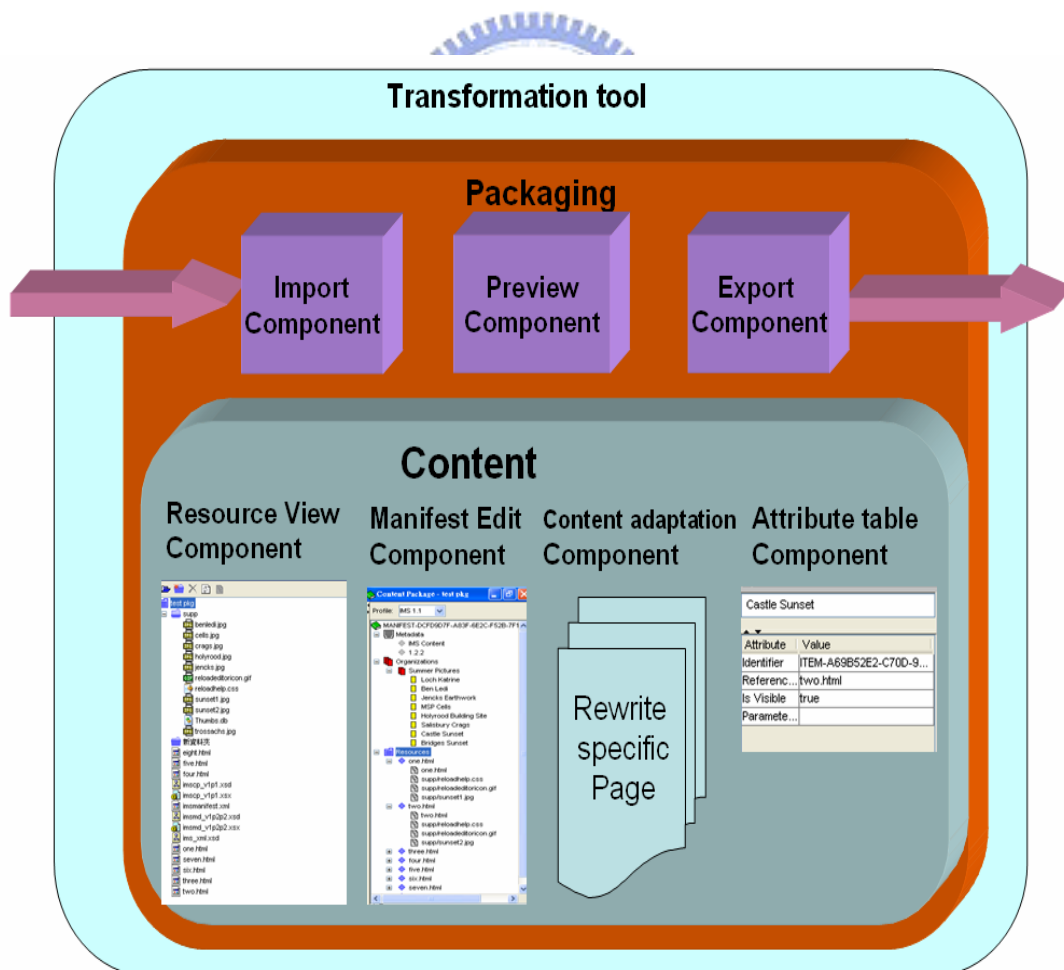


Fig 3-0-1 Toolkit architecture of our system.

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 3-1. In this chapter, each part of the architecture will be detail described.

### 3.3 Toolkit composition

Figure3-2 and Figure3-3 show the workflow of our processes.

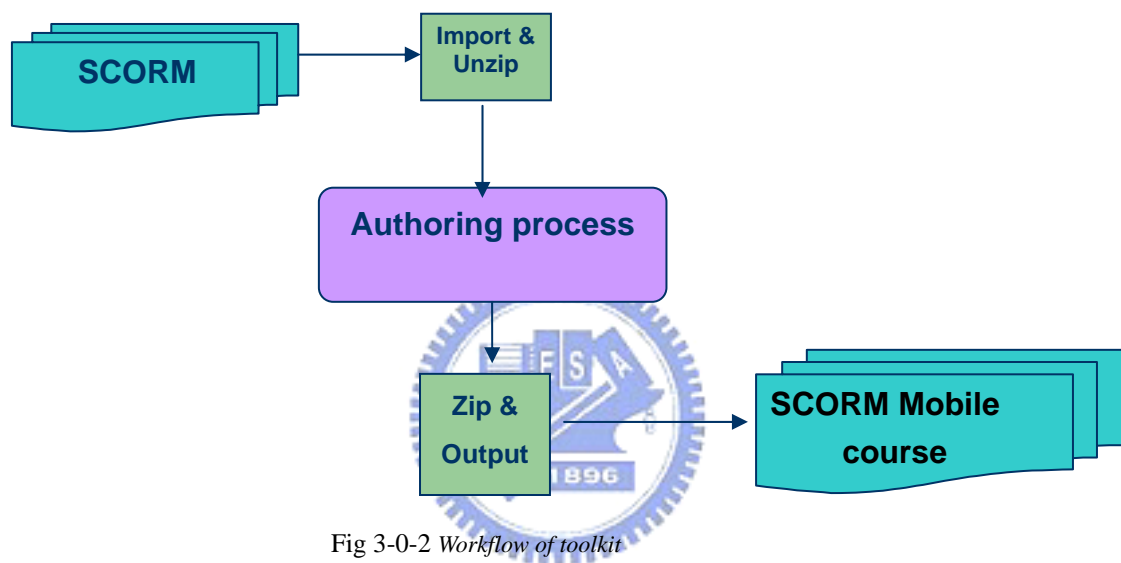


Fig 3-0-2 Workflow of toolkit

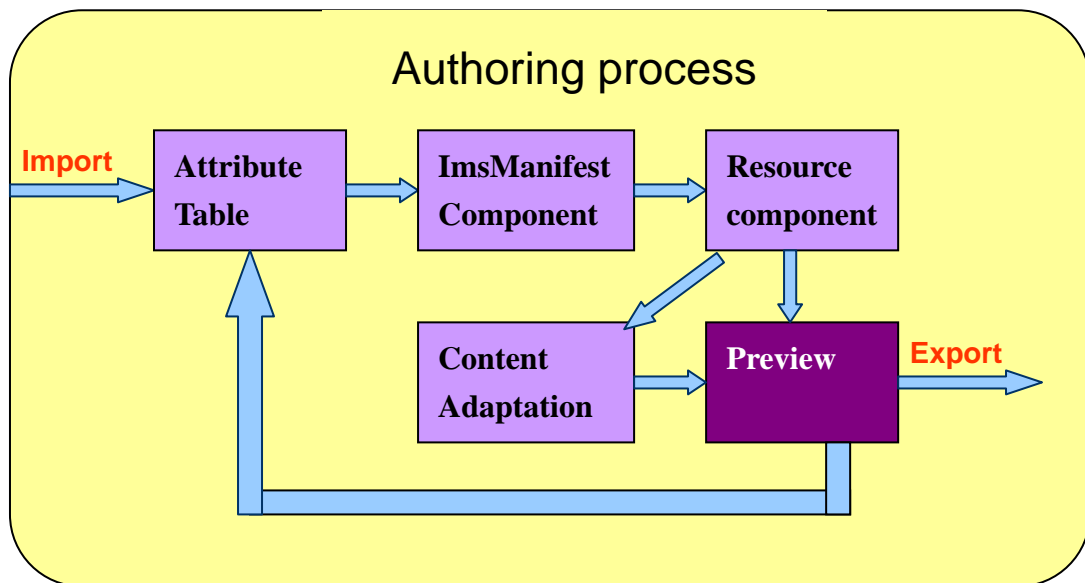


Fig 3-0-3 Toolkit authoring process

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 Figure3-4, 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 Figure3-5 and Figure3-6, it means we divided original page as show in Figure 3-4 into two pages.

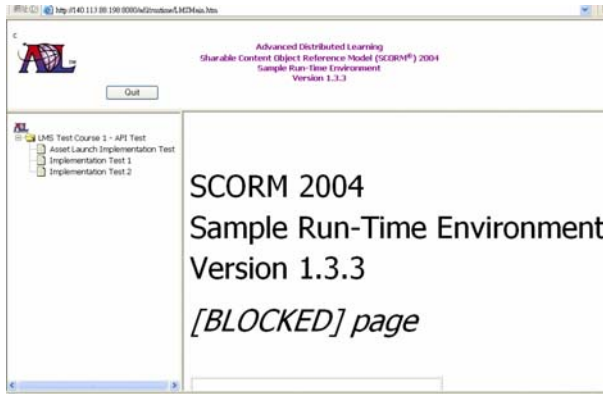


Fig 3-0-4 The original page of SCORM SRTE

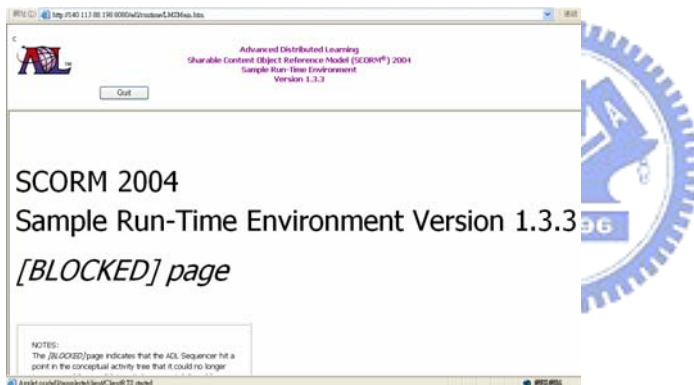


Fig 3-0-5 The snapshot after reduce left navigate page

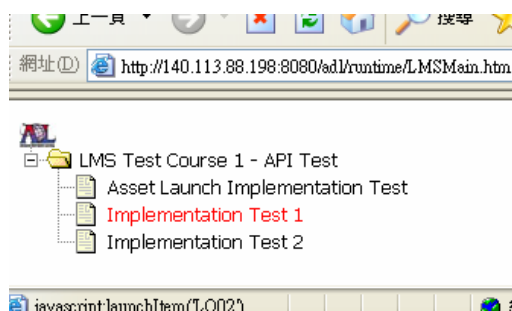


Fig 3-0-6 The navigation of SCORM

### 3.3.2 Resource View Component

Resource view component response for individual file level handling. In order to

cooperate with Figure 3-5 and 3-6, 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

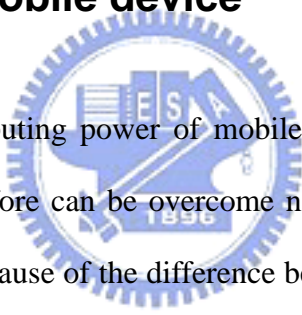
SCORM specification does not define what kind of source file it should be, it only provides for the rule for communication and appear sequence. Therefore there are too many possibility that source will be. As the document presentation format analysis is chapter2.5, some document format can not be transformed 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 was 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 the JavaScript will work 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.



---

# Chapter 4 Implementation

---

## 4.1 Developing environment – “Reload Editor”

Due to SCORM specification is complex and needs much time to implementation all of the mechanism. We use an existing open source software “Reload editor” [7] as show in figure10. This software is implementation by Java language. We use their document structure handler and parser to accomplish our mechanism.

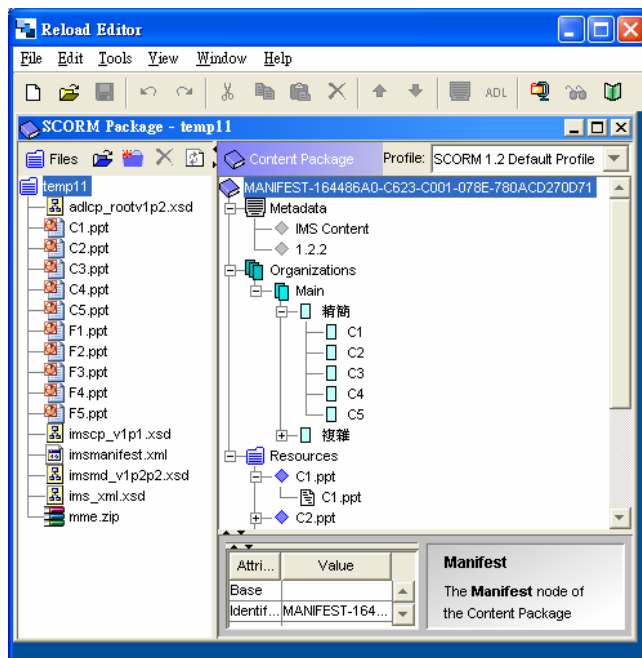


Fig 4-0-1 Snapshot of Reload Editor.

## 4.2 System view

We use wireless802.11 to keeping the connection. In this way, we do not care about the connection problems. The Figure 4-2. and Figure 4-3. are the snapshot of our mobile device toolkit.

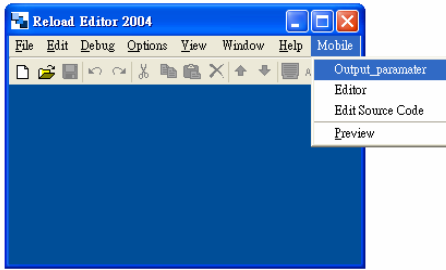


Fig 4-0-2 The plug-in snapshot of our toolkit

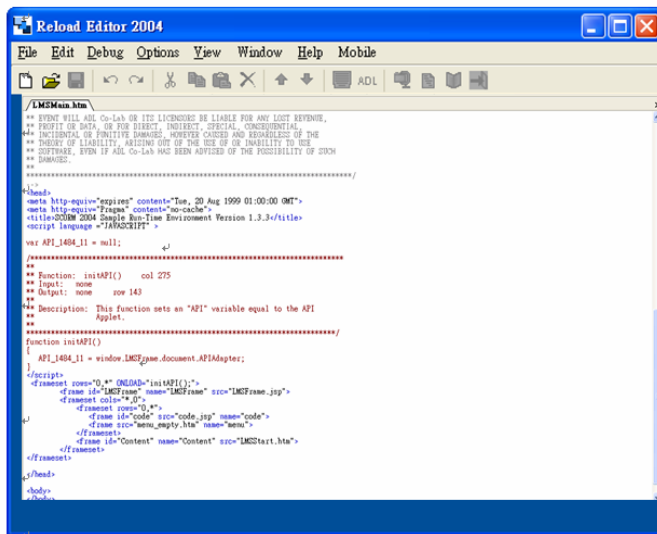


Fig 4-0-3 The snapshot of source code editing function

## 4.2.1 Resource view component modification

In resource view component, we add three links to html document, they are:

1. We add a hyperlink (Top) to return top.html page. The html code is as follow:

```
<P align="right"><a href="top.html">Top</a></P><P>&nbsp;</P>
```

2. Add a JavaScript control code in first page of every SCO

```
SetValue("adl.nav.request", "continue")
```

3. Add a JavaScript control code in last page of every SCO

```
SetValue("adl.nav.request", "previous")
```

As show in Figure4-4, we use a SCORM 2004 data model function ” ---adl.nav.request ” navigate browsing. This function enable browse from one SCO to another SCO.

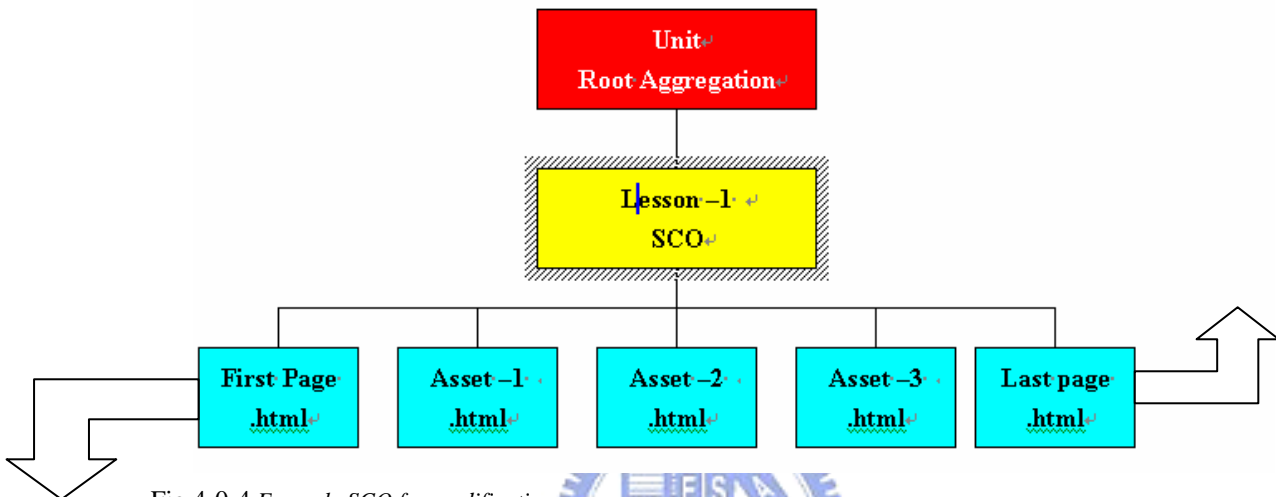


Fig 4-0-4 Example SCO for modification



### 4.3 Rte view

As the picture show in Figure 4-4, we create a page “top.html” represent the document structure and combine with Assets. In Fiure4-5, we add a link “Top” at left side of document. There are no existing navigational frames, in order to browse this teaching material, the additional link is necessary. Although this link decrease browsing convince, but save space.



Fig 4-0-5 *top.html* view

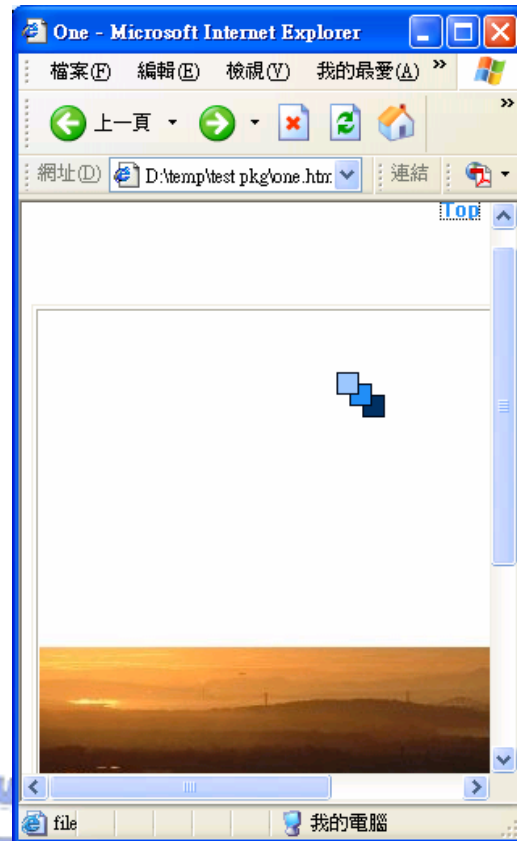


Fig 4-6 Asset View

## 4.4 Discussion

In this chapter, we had demonstrated our transformation tool and sample result. However, there are many existing problems while implement this tool. For example, we want to design this tool and browsing on pocket PC. But there comes the problem that the pocket IE do not support the JavaScript invoke Java Applet well. They either only support Java Applet function or JavaScript function partially. Other research implement their own RTE and framework to overcome this problem, but we do not want to build up our own RTE due to the limitation of time. We believe that this problem will be solved soon due to the advancement of software version support. The other problem is sometimes when we export SCORM document to another LMS RTE,

our document can not adjust to their configuration. Maybe this is because there are some conflict setting between “Reload Editor” and the RTE.



---

## Chapter 5 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.



### 5.1 Conclusion

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.

## 5.2 Future work

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.





---

## Chapter 6 Reference

---

- [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] 許政穆、許健飛、鐘國麟。 “Design of the SCORM-based e-Learning  
Environment for Mobile Carriers”。 WISCS 2003 網路教學系統平台與內容標  
準化」學術研討會。 2003。 高雄。 896
- [6] 張玄善, 林慧珍, “行動裝置上的 SCORM 教材瀏覽器” 淡江大學碩士班論  
文 民國 94 年
- [7] Reload editor <http://www.reload.ac.uk>
- [8] SRTE <http://www.adlnet.gov/scorm>
- [9] SCORM Player <http://www.reload.ac.uk/tools.html>
- [10] Carnegie Mellon University, Learning Systems Architecture Lab., SCORM  
Best Practices Guide for Content Developers, <http://www.lsal.cmu.edu>
- [11] <http://www.scormexplorer.com/catalog.asp?cate=7>
- [12] Opera8.5 <http://www.opera.com>
- [13] TK Shih, NH Liit, HP Chang, KH Huang, Adaptive Pocket SCORM Reader  
Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on,

2004

- [14] I. Waycott, and A. Kuhlska-Hulme (?003), "Students' experiences with PDAs for reading course materials" Personal and Ubiquitous Computing Volume 7, Issue 1 (May 2003), ISSN:1617-4909
- [15] Kay Sommers, Jane Hcsler, and Jim Bastick (ZOOI), "Little Guys Make a Big Splash PDA Projects at Virginia Commonwealth University", Proceedings of the 29th annual ACM SIGUCCS conference on User services (SIGUCCS' 01)
- [16] 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
- [17] 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
- [18] 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