

國立交通大學

資訊科學系

碩士論文

適應性內容傳輸機制於多樣行動載具之研究



An Adaptive Content Delivery Mechanism for
Multiple Mobile Devices

研究生：陳君翰

指導教授：曾憲雄 教授

中華民國九十四年六月

適應性內容傳輸機制於多樣行動載具之研究
An Adaptive Content Delivery Mechanism for Multiple Mobile Devices

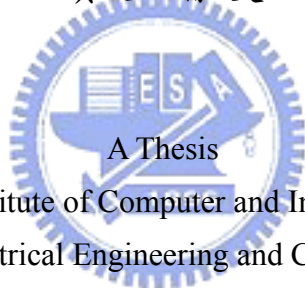
研究生：陳君翰
指導教授：曾憲雄

Student : Chun-Han Chen
Advisor : Shian-Shyong Tseng

國立交通大學

資訊科學系

碩士論文



Submitted to Institute of Computer and Information Science
College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Science

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

適應性內容傳輸機制於多樣行動載具之研究

研究生：陳君翰

指導教授：曾憲雄 博士

國立交通大學電機資訊學院

資訊科學系

摘要

隨著行動載具的性能越來越強，在無線網路的環境之下傳輸合適的學習資源給多樣化行動載具的需求也越來越大。但是因載具能力因素以及頻寬不足的情形，學習內容本身無法展現出令人滿意的效果，因此如何將學習資源做內容適應性轉換形成一個重要課題。先前學習者取得學習物件時，當時的網路情況與硬體、使用者偏好設定將會被記錄下來，這提供了我們解決問題的靈感，當新使用者要求學習物件時，可以依據和先前學習者的設定偏好相似程度，先行給予最接近需求的內容，不但可同時兼顧內容傳輸上的效能，也能夠兼顧使用者對於學習內容的滿意程度。在本論文中，我們提出了適應性內容傳輸機制(ACDM)於符合SCORM 教學平台的元件資料庫(LOR)中。ACDM 可以有效率地管理先前學習者的傳輸記錄，快速判斷目前學習者及以往記錄中設定的相似程度，以送出更符合使用者本身需求的學習內容。ACDM 包含了資料格式定義階段，與內容傳輸階段，前者依據硬體、使用者設定與頻寬狀態定義了內容轉換規則(CAR)，而後者則包含了內容轉換管理機制(CAMS)，將 CAR 實行群集以及決策樹分析(CADT)，也包含了適應內容決策進程(ADP)以根據決策樹判斷何內容版本符合當前使用者需求；最後包括內容合成機制(CS)，負責將單一媒體根據參數轉換成不同的版本。最後的實驗結果展現出 ACDM 機制在內容傳輸上有明顯的效果改善。

An Adaptive Content Delivery Mechanism for Multiple Mobile Devices

Student: Chun-Han Chen

Advisor: Dr. Shian-Shyong Tseng

Institute of Computer and Information Science

National Chiao Tung University

Abstract

With the mobile devices becoming more and more powerful, the requirements of delivering and displaying the learning contents on the diverse mobile devices over the wireless network are increasing rapidly. However, the insufficient hardware capabilities and the limited bandwidth have led to the bad navigation experience and unfavorable presentation of contents. Therefore, how to perform the *Content Adaptation* becomes an important issue. The historical user's request records including hardware capabilities, user's preference, and the current network situation, can provide us an opportunity to solve the above issue. The inspired concept is that if we can provide a new user's request with the adaptive contents created from previous similar user's request, not only the performance of content delivery but also the user's satisfaction will be improved highly.

Therefore, in this thesis, based on the aforementioned concept and SCORM compliant learning object repository (LOR), we propose an **Adaptive Content Delivery Mechanism**, called **ACDM**, which can efficiently manage a large number

of historical user's requests, and intelligently deliver a proper adaptive content with higher fidelity from LOR to the user directly and then prepare a transcoded content version for next similar request. The ACDM includes **Adaptation Data Format Definition Phase** and **Adaptive Content Delivery Phase**. The former defines an adaptation data format, called **Content Adaptation Rule (CAR)**, based upon CC/PP, UAProf, etc. In order to efficiently deliver the suitable content with associated learning resources to users in accordance with their user preferences, hardware capabilities, and variable wireless bandwidth, the latter consists of **1) Content Adaptation Management Scheme (CAMS)**: apply clustering approach and decision tree approach successively to create a **Content Adaptation Decision Tree (CADT)**, which can be used to decide the appropriate adaptive contents from the LOR, **2) Adaptation Decision Process (ADP)**: propose an **Adaptation Decision Process Algorithm** to intelligently determine a suitable version of the existing adaptive content based on the **CADT**, and **3) Content Synthesizer**: transcode the content if necessary. For evaluating our proposed approach, an ACDM prototypical system is developed. Furthermore, the experimental results show that the ACDM is workable and beneficial.

Keywords: **Mobile learning, content adaptation, content delivery.**

誌謝

首先感謝指導教授曾憲雄博士，從大學時代至研究所四年來在求學及處世等各方面的指導與啟發，使我獲得許多寶貴經驗，受益匪淺，感激不盡。同時也感謝我的口試委員，黃國楨教授、楊鎮華教授與楊錦潭教授，他們給予了我相當多寶貴意見，讓這篇論文更有價值。

感謝博士班蘇俊銘學長、翁瑞鋒學長，兩年的指導期間，讓我學會了許多理論知識與實務技巧，也給予了我的論文許多寶貴的意見，解答了我的疑惑，協助我論文的修改工作，深表感激。

同時也感謝實驗室其他的同窗夥伴，宋昱璋、林易虹、黃柏智、李育松、吳政霖、陳瑞言與邱成樑，不管是學業上或是生活上，兩年下來的彼此互助與幫忙，一起渡過了既忙碌、又充實的碩士生涯。

最後，我要感謝我的家人，和女朋友薇艷對我的支持與鼓勵，讓我在面對挫折的時候能夠繼續向前，也讓我能夠有相當的自信完成此篇論文。

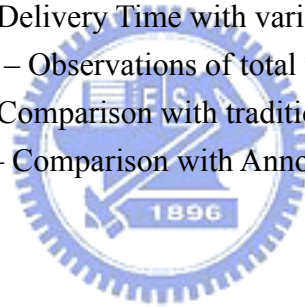
謝謝每一位曾經幫助我的人。

Table of Contents

摘要.....	i
Abstract.....	ii
誌謝.....	iv
Table of Contents	v
List of Tables.....	vii
List of Algorithms	vii
Chapter 1 Introduction.....	1
Chapter 2 Related Work.....	4
2.1 SCORM (Sharable Content Object Reference Model)	4
2.2 Other Related Research.....	6
Chapter 3 Adaptive Content Delivery Mechanism (ACDM).....	9
3.1 The Framework of Adaptive Content Delivery Mechanism (ACDM).....	9
3.2 Adaptation Data Format Definition.....	11
Chapter 4 Content Adaptation Management Scheme (CAMS).....	19
4.1 The Clustering Process of CAMS.....	19
4.2 Decision Tree Construction.....	24
4.3 CADT Maintain Process.....	25
Chapter 5 Adaptive Content Delivering Process	27
5.1 Content Adaptation Process.....	27
5.2 Adaptation Decision Process (ADP)	30
5.3 Content Synthesizer.....	33
Chapter 6 Implementation and Experimental Results.....	35
6.1 The System of ACDM	35
6.2 Experiments.....	39
Chapter 7 Conclusions.....	43
Bibliography	44

List of Figures

Figure 2.1: SCORM Content Packaging Scope and Corresponding Structure of Teaching Materials.....	5
Figure 3.1: The Architecture of Adaptive Content Delivery Mechanism (ACDM).....	10
Figure 4.1: The Process of Content Adaptation Management Scheme (CAMS).....	19
Figure 4.2: The Content Adaptation Decision Tree (CADT) based upon the Hardware Profiles (HP) in Table 3.3.....	25
Figure 6.1: Main flow for a user to retrieve a learning object.....	35
Figure 6.2: Content Adaptation.....	36
Figure 6.3: Preference-based Transformation.....	37
Figure 6.4: Progressive Adaptation.....	38
Figure 6.5: ACDM Web Server Monitoring	39
Figure 6.7: Exp 1 – Delivery Time with various bandwidth settings	40
Figure 6.8: Exp 1-B – Observations of total transmission size.....	40
Figure 6.9: Exp 2 – Comparison with traditional web server.....	41
Figure 6.10: Exp 3 – Comparison with Annotation-based Server	42



List of Tables

Table 3.1: The Hardware Profile (HP) of Mobile Device.....	13
Table 3.2: The Description of User Preference (UP) Attributes	15
Table 3.3: An Example of CAR for the same page (p_j) in a Learning Object (LO_i).....	18
Table 4.1: The Result of Applying UP Clustering Algorithm	24
Table 4.2: The Result of Mapping the Numerical Value in HP	25
Table 6.1: Dynamic bandwidth re-estimation.....	42

List of Algorithms

Algorithm 4.1: UP Clustering Algorithm (UPCALgo).....	23
Algorithm 5.1: Content Adaptation Process (CAP) Algorithm (CAPALgo)	29
Algorithm 5.2: Adaptation Decision Process Algorithm (ADPALgo)	32
Algorithm 5.3: Adapted Content Generation.....	34

Chapter 1

Introduction

With the rapid development of the Internet, e-learning system has become more and more popular because it can make users conveniently study at any time and any location. In addition, with the mobile devices, such as PDA, Cell Phone, Pocket PC, etc., become more and more powerful, the requirements of delivering and displaying the learning contents on the diverse mobile devices over the wireless network are increasing rapidly. However, the insufficient hardware capabilities, e.g., memory and screen size, and the limited bandwidth of wireless network have led to the bad navigation experience and unfavorable presentation of learning contents. Therefore, how to transform the existing learning contents into a suitable version, which can be efficiently delivered and displayed to meet the users' needs and mobile devices, called *Content Adaptation*, becomes an important issue. Currently, to solve the above issue, many articles have proposed several content adaptation approaches including the web structure analysis [XWW04] [CWZ03] [GF+02], information summarization [OOH02] [OHA01], pre-annotation process [VLC04] [HMZ04] [TNS03], and context-aware decision [LWF03]. These approaches employ either the transcoding [IMAGICK][SOX] or content caching approach. Nevertheless, real time transcoding technique leads to a long latency and response time. Although the content caching approach can decrease the transcoding latency, the cached content may not meet the user needs.

The historical user's request records including hardware capabilities, user's preference, and the current situation of wireless network, can provide us an opportunity to solve the above issues. Accordingly, the inspired concept is that if we

can provide a new user's request with the adaptive contents created from previous similar user's request, not only the performance of content delivery but also the user's satisfaction will be improved highly. Furthermore, Sharable Content Object Reference Model (SCORM) 2004 [SCORM], the most popular standard on e-learning system, proposes the Content Packaging scheme to package the associated learning resources, described by the metadata into a learning course, which is usually stored in the Learning Object Repository (LOR). Therefore, how to efficiently manage a large number of historical user's requests and deliver the suitable SCORM compliant learning contents to mobile device is our concern.

Therefore, in this thesis, to consider not only the hardware capability of mobile device and user's preference, but also the dynamic bandwidth, based on the aforementioned concept and SCORM compliant learning object repository (LOR), we propose an **Adaptive Content Delivery Mechanism**, called **ACDM**, which can efficiently manage a large number of historical user's requests, and intelligently deliver a proper adaptive content with higher fidelity from LOR to the user directly and then prepare a transcoded content version for next similar request. Thus, the **ACDM** includes two phases: **Adaptation Data Format Definition Phase** and **Adaptive Content Delivery Phase**. The former defines an adaptation data format including the **User Preference (UP)**, **Hardware Profile (HP)**, and **Media Parameter (MP)** based upon Composite Capability/Preference Profiles (CC/PP) [CCPP], User Agent Profile (UAProf) [UAPROF], and FIPA Device Ontology Specification [FIPA02]. These adaptation data format can be used to indicate the useful information about the user, hardware, and media during content adaptation and delivery. In order to efficiently deliver the suitable content with associated learning resources to users in accordance with their user preferences, hardware capabilities, and variable wireless

bandwidth, the latter consists of three modules, **1) Content Adaptation Management Scheme (CAMS):** apply the distance based clustering approach to group the historical user's requests into several groups according to the User Preference (UP) and then apply decision tree approach to create a **Content Adaptation Decision Tree (CADT)**, which can be used to efficiently decide the appropriate adaptive contents from the LOR, **2) Adaptation Decision Process (ADP):** propose an **Adaptation Decision Process Algorithm**, called **ADPAlgo**, which can decide a suitable version of the existing adaptive content based on the **CADT**, and **3) Content Synthesizer:** use the decided adaptation parameters to transcode the content if necessary according to the results of ADPAlgo. Finally, for evaluating our proposed approach, a prototypical system of ACDM based on SCORM compliant learning object repository (LOR) is developed. Some experiments have also been done. The experimental results show that the ACDM is workable and beneficial.



Chapter 2

Related Work

In this section, we review SCORM standard and some related work as follows.

2.1 SCORM (Sharable Content Object Reference Model)

Among those existing standards for learning contents, SCORM [SCORM], which is proposed by the U.S. Department of Defense's Advanced Distributed Learning (ADL) organization in 1997, is currently the most popular one. The SCORM specifications are a composite of several specifications developed by international standards organizations, including the IEEE [LTSC], IMS [IMS], AICC [AICC] and ARIADNE [ARIADNE]. In a nutshell, SCORM is a set of specifications for developing, packaging and delivering high-quality education and training materials whenever and wherever they are needed. SCORM-compliant courses leverage course development investments by ensuring that compliant courses are "RAID:" Reusable: easily modified and used by different development tools, Accessible: can be searched and made available as needed by both learners and content developers, Interoperable: operates across a wide variety of hardware, operating systems and web browsers, and Durable: does not require significant modifications with new versions of system software.

In SCORM, content packaging scheme is proposed to package the learning objects into standard teaching materials, as shown in Figure 1. The content packaging scheme defines a teaching materials package consisting of 4 parts, that is, **1) Metadata:** describes the characteristic or attribute of this learning content, **2)**

Organizations: describe the structure of this teaching material, **3) Resources:** denote the physical file linked by each learning object within the teaching material, and **4) (Sub) Manifest:** describes this teaching material is consisting of itself and another teaching material. In Figure 1, the organizations define the structure of whole teaching material, which consists of many organizations containing arbitrary number of tags, called *item*, to denote the corresponding chapter, section, or subsection within physical teaching material. Each item as a learning activity can be also tagged with activity metadata which can be used to easily reuse and discover within a content repository or similar system and to provide descriptive information about the activity. Hence, based upon the concept of learning object and SCORM content packaging scheme, the teaching materials can be constructed dynamically by organizing the learning objects according to the learning strategies, students' learning aptitudes, and the evaluation results. Thus, the individualized teaching materials can be offered to each student for learning, and then the teaching material can be reused, shared, and recombined.

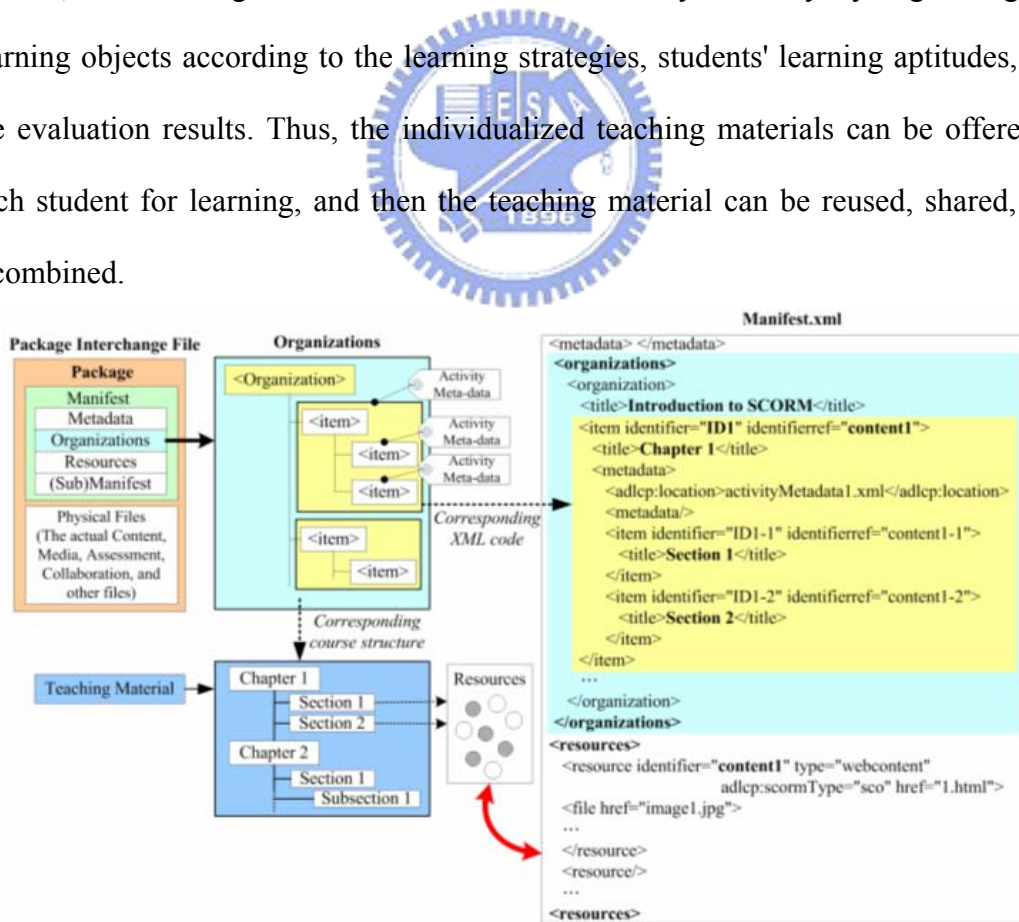


Figure 2.1: SCORM Content Packaging Scope and Corresponding Structure of Teaching Materials

2.2 Other Related Research

With the requirements of rendering the learning contents deployed on web server on mobile devices, the diverse content adaptation approaches have been proposed. The simplest way is to utilize the native content adaptation functionalities of browsers of user's handheld device [HFTS][PWB3C]. Currently, most browsers of mobile devices can automatically resize the web pages into the suitable small ones to fit the mobile screen size and try to eliminate the horizontal scroll bar to facilitate the web navigation. Obviously, this content adaptation mobile browsers support is too simple to meet the diverse handheld devices much less the user needs.

Although pre-authoring multiple versions of a content to fit variable mobile devices is a workable approach, it is very time consuming and cumbersome for content providers. Accordingly, many dynamic content adaptation approaches including **Web Structural Analysis**, **Information Summarization**, **Pre-Annotation Process**, and **Context-aware Decision** have been proposed to solve above issues.

Lemloulmal and Layaida [TNS03] proposed a framework to deliver adapted content for different target contexts based on a Universal Profiling Schema (UPS) describing the environment characteristics and a profile exchange protocol. In their system, the descriptions of the client context, the server capabilities, the document profile, etc. are pre-annotated first. Thus, a negotiation strategy is used to find a suitable content variant according to the server capabilities and the client preferences. Then, a dynamic adaptation approach employs the XSLT [XSLT] and transcoding approach to generate an appropriate adapted content variant. Hinz et al. [HMZ04] proposed a system, called AMACONT, which defines an XML-based user model consisting of Device, Identification, and Preference profiles based on CC/PP

(Composite Capability / Preference Profiles) and an RDF grammar. AMACONT includes Static Adaptation and Automatic Layout Adaptation processes. The former generates a number of variants for each document component and the latter can attach an XML-based layout descriptions to describe the presentation of component-based Web documents. By the similar approach, Villard et al. [VLC04] also proposed the specification of a document presentation model to adapt the content for generic document production. Their presentation process also employs the XSLT approach to perform document adaptation according to the related constraint. However, their approaches stated above perform the content adaptation relying on pre-annotating all involved contents or documents, which results in the complex preprocessing and limits the flexibility and scalability. Besides, they could not take into account the user's preference and situation of wireless network well.

For navigation on small display devices, a Hierarchical Atomic Navigation Concept, called HANd, was proposed by Gonzalez-Castano et al. [GF+02]. In HANd, a *navigator page* generated automatically is used to indicate some or all elements embedded in a WWW page. Therefore, in order to generate the navigator page, a web page has to be analyzed and fragmented into several separated "clipped" versions, which can be delivered to a small display device, according to an importance value for every page fragment. Here, low importance fragments are ignored when display space is limited. Similarly, Chen et al. also proposed a browsing convention and adaptation scheme to facilitate the navigation on mobile devices. Their approach organizes a web page into a two level hierarchy, where the top level with a thumbnail representation like a *navigator page* in [GF+02] for providing a global view and the bottom level with index to a set of sub-pages for detail information. Then, based on the concept of web structure analysis, Yin and Lee [YL04] focuses on the important parts of web pages to propose a ranking algorithm in order to reduce the number of

page fragments and delivery latency. Although the web structural analysis approaches proposed in [GF+02] [CWZ03] [YL04] can improve the delivery latency, the quality of content shown on mobile devices are limited on fixed page fragments and the user's preference is not considered yet.

Buyukkokten et al. [OHA01] [OOH02] utilized the concept of information summarization to summarize the parts of Web pages and HTML forms. In this approach, each web page is broken into several text units of which each can be hidden, partially displayed, made fully visible, or summarized. Besides, they provided five methods, *Incremental*, *All*, *Keywords*, *Summary*, and *Keyword/Summary*, to control the navigation behaviors. However, not all web pages are suitable to do text summarization and the summarized statements as losy information may mislead users.

Furthermore, in order to consider the user's preference, Lum and Lau [LL03] proposed a decision engine to automatically determine an appropriate content adaptation version based on QoS-sensitive approach. Therefore, a typical score tree with several score nodes is created to be used for evaluating the QoS of the content versions in various quality domains. Nevertheless, the predefined score tree with limited content version may not meet the users' needs and may constrain the flexibility and extensibility of system.

Chapter 3

Adaptive Content Delivery Mechanism (ACDM)

As mentioned above, with the rapid development of network technology and mobile device, how to efficiently provide a user with a suitable content according to the user's hardware capability of mobile device, preference, and unstable bandwidth over the wireless environment has become an important issue. Therefore, in this thesis, based on the concept of providing a user with the adaptive content created from the previous similar request, how to efficiently manage these historical user's requests are our concern. Therefore, the following three issues have to be solved: (1) how to define the useful data format to present the user's request, (2) how to group several user's requests with the similar user preferences into the same group, and (3) how to decide the suitable adaptive content with high fidelity for a new user's request. Accordingly, we propose an **Adaptive Content Delivery Mechanism**, called **ACDM**, in this chapter.

3.1 The Framework of Adaptive Content Delivery Mechanism (ACDM)

The ACDM can efficiently manage a large number of historical user's requests, and intelligently determine to deliver an existing adaptive content with higher fidelity from LOR to the user directly and then prepare a transcoded content version for next similar request. The ACDM Architecture is shown in Figure 3.1.

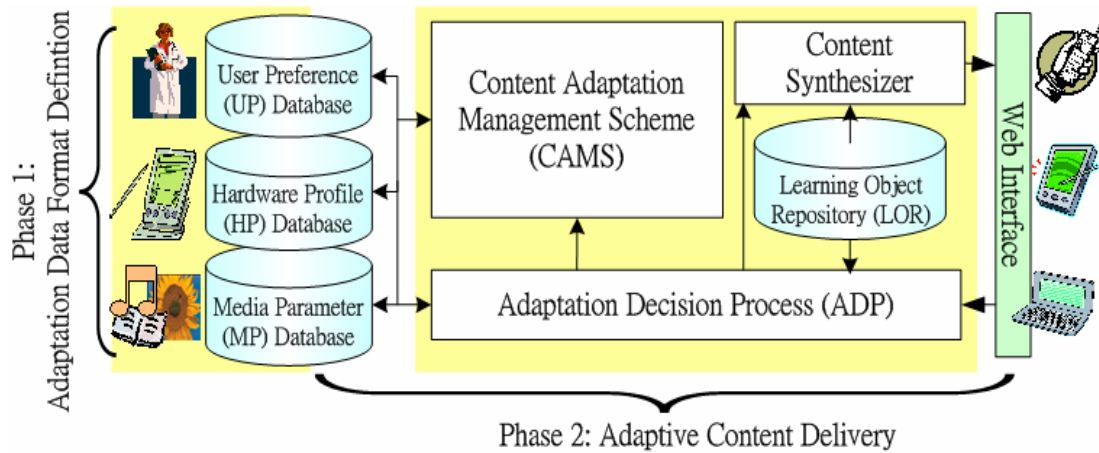


Figure 3.1: The Architecture of Adaptive Content Delivery Mechanism (ACDM)

The ACDM includes two phases described as follows:

1. **Adaptation Data Format Definition Phase:** we define firstly the adaptation data format including **User Preference (UP)**, **Hardware Profile (HP)**, and **Media Parameter (MP)** based upon Composite Capability/Preference Profiles (CC/PP) [CCPP], User Agent Profile (UAProf) [UAPROF], and FIPA Device Ontology Specification [FIPA02]. Here, the **UP** describes the desired preference of a user, e.g., the desired delivery time, image format, the ratio of audio to picture, etc., the **HP** describes the hardware capabilities of user's mobile device, e.g., the device type, screen size, etc, and the **MP** describes how to transcode the image and audio within a requested learning objects.
2. **Adaptive Content Delivery Phase:** for efficiently delivering the suitable content with associated learning resources to users in accordance with their user preferences, hardware capabilities, and variable wireless bandwidth, we propose the following three modules.
 - **Content Adaptation Management Scheme (CAMS):** for efficiently managing the historical user's requests data, we firstly apply the distance based

clustering approach to group the historical user's requests into several groups according to the User Preference (UP). After the clustering phase, every created cluster with similar user preference will be tagged with a cluster labels. Thus, the Hardware Profiles (HP) within the history user requests with corresponding cluster label are used as training data to create a decision tree, called **Content Adaptation Decision Tree (CADT)**. The CADT thus can be used to efficiently decide the appropriate adaptive contents.

- **Adaptation Decision Process (ADP):** for intelligently determining a suitable adaptive content version based on CADT, we propose a **Adaptation Decision Process Algorithm**, called **ADPAlgo**, which can decide a suitable version of the existing adaptive content.
- **Content Synthesizer:** according to the results of ADPAlgo, the content synthesizer will use the decided adaptation parameters to transcode the content if necessary.



The details of each phase will be described in the following chapters.

3.2 Adaptation Data Format Definition

For efficiently manage the existing user's requests, we have to define a data format including the User Preference (UP), Hardware Profile (HP), and Media Parameter (MP), which will be recorded in database, to represent every user's request based upon CC/PP [CCPP], UAProf [UAPROF], and Device Ontology Specification [FIPA02]. These definitions will be described in this chapter.

As mentioned in Chapter 2, a SCORM compliant Learning Object (LO) usually

consists of several content pages with associated learning resources. Therefore, a Learning Object (LO) can be defined as follows:

Definition 1: Learning Object (LO), $LO = (P, R)$, where

- $P = \{p_1, p_2, \dots, p_n\}$: denotes the related pages in an LO.
- $R = \{r_1, r_2, \dots, r_m\}$: denotes the related learning resources, e.g., HTML, images, and audio.

In this thesis, we assume that one user's request will acquire a page consisting of an HTML file, several images, and a background audio only, which can be shown by browser at the same time. Consequently, a $p_j \in P$ can be represented as $p_j = (HTML, r_1, r_2, \dots, r_k)$, where $j \leq n$, $k \leq m$. Thus, the LO_{ij} can denote the j th page (P_j) of the i th LO (LO_i) in a SCORM compliant learning object repository (LOR).

Then, Table 3.1 shows the attributes with associated value in Hardware Profile (HP). We define the general value of every attribute to describe the diverse mobile device. In addition, the Symbol * in Table 3.1 denotes this attribute's value can not be automatic detected by UAProf and HTTP request header during the content delivery. Thus, the HP is defined in Definition 2.

Table 3.1: The Hardware Profile (HP) of Mobile Device

Attribute	Description and Related Value
Machine Type (MT)	0: Cell Phone, 1: Cell Phone/PDA, 2: Palm, 3: Pocket PC, 4: Laptop
* CPU Speed (CPU)	The value within the range from 66 to 500 .
* System Memory (SM)	The value within the range from 8 to 1024 .
Screen Horizontal Size (SHS)	The value within the range from 40 to 800 .
Screen Vertical Size (SVS)	The value within the range from 40 to 800 .
Color Depth (CD)	2: 2 bits, 4: 4 bits, 8: 8 bits, 16: 16 bits
Sound Precision (SP)	U: Unsupported, 8: 8 bits, 16: 16 bits
Sound Rate (SR)	U: Unsupported, 4: 4 KHz, 8: 8 KHz, 11: 11 KHz, 22: 22 KHz, 44: 44KHz
Sound Channels (SC)	U: Unsupported, M: Mono, S: Stereo
Networking (Net)	Ir: IrDA, WL: Wireless LAN, GSM: GSM, GPRS: GPRS, BT: Bluetooth



Definition 2: Hardware Profile (HP) is as a vector with related hardware attributes.

- **HP** = <**MT, CPU, SM, SHS, SVS, CD, SP, SR, SC, Net**>, where every attribute denotes a specific capability of mobile device. For example, the **MT** denotes that the Machine Type of device is cell phone without or with PDA function (*Cell Phone/PDA*), or palm.

Furthermore, in order to deliver the suitable content to meet the user's needs over wireless environment, we take not only the associated resources in a page of an LO, but also the delivery time depending on the bandwidth into account to represent the **User Preference (UP)**. Table 3.2 describes the UP's attributes and associated values. Here, the *Delivery Time (DT)* and *Ratio* attributes have the maximum value which will be used to estimate the similarity between two UPs in **Adaptive Content Delivery Phase**, described in Chapter 4. Regarding the quality of associated pictures and audios in a page, users can use these attributes, **PPO, PFO, and APO**, to define the preferred priority of quality factors. For example, the string **JPG** denotes that the user prefer the JPG picture format to the PNG.

Moreover, in general, the pictures embedded in a page can be divided into 2 categories: *Foreground* and *Background pictures*. In ACDM, for offering user to flexibly control the appearance of images, we define two additional image types, *Icon*, and *Bar*, which can be identified automatically according to their ratio of width to height from the *Foreground pictures*. User can thus use the switch attributes of UP to control whether showing it or not. In order to facilitate the UP setting process for users, the supported image and audio formats can also be automatically indicated from the information in UAProf [UAPROF] and Agent Header [RFC2616-14]. In other words, the setting of image and audio will be automatic disabled, if it can not be supported by user's handheld device.

Table 3.2: The Description of User Preference (UP) Attributes

Attribute		Description and Related Value
Delivery Time (DT)		The maximum delivery time which the user may tolerate. The default maximum value is 30 seconds.
Ratio		The expected ratio of audio to picture . The default maximum ratio is 1:20.
Picture Ordering (PPO)	Property	The order of three properties (D : <i>Dimension</i> , C : <i>Color Depth</i> , Q : <i>Quality</i>). For example, the string DCQ denotes that the priority order is $D > C > Q$.
Picture Ordering (PFO)	Format	The order of four formats (J : <i>JPG</i> , P : <i>PNG</i> , G : <i>GIF</i> , B : <i>BMP</i>). For example, the string JPGB denotes that the priority order is $J > P > G > B$.
Audio Ordering (APO)	Property	The order of three properties (S : <i>Samples Rates</i> , B : <i>Bits</i> , C : <i>Channels</i>). For example, the string SBC denotes that the priority order is $S > B > C$.
Foreground Switch (FPS)	Picture	Show the foreground picture except <i>Icon</i> and <i>Bar</i> or not (0 : <i>false</i> , 1 : <i>true</i>).
Background Switch (BPS)	Picture	Show the background picture or not (0 : <i>false</i> , 1 : <i>true</i>).
Icon Switch (IS)		Show the icon picture or not (0 : <i>false</i> , 1 : <i>true</i>).
Bar Switch (BS)		Show the picture of separation line or not (0 : <i>false</i> , 1 : <i>true</i>).
Audio Switch (AS)		Play the background audio or not (0 : <i>false</i> , 1 : <i>true</i>).

Accordingly, UP can be defined as follows:

Definition 3: User Preference (UP) is as a vector.

- $UP = \langle DT, Ratio, PPO, PFO, APO, FPS, BPS, IS, BS, AS \rangle$, where every attribute denotes a specific requirement of a user.

Therefore, based upon the **LO**, **HP**, and **UP**, a new **User Request (UR)** can be defined as $UR = (LO, (B, HP, UP))$. In addition, in ACDM, the media transcoding process will be triggered if there is no suitable adaptive content for a new UR. The transcoding process will employ the transcoding tools [IMAGICK][SOX] to transform the media format into different type and file size depending on given transcoding parameters, e.g., the value of picture properties (**D**: *Dimension*, **C**: *Color Depth*, **Q**: *Quality*) or audio properties (**S**: *Samples Rates*, **B**: *Bits*, **C**: *Channels*). Therefore, for a new user's request, every media resource will be associated with a **Media Parameter (MP)** including media type (**T**), resultant transcoded size (**S**), and transcoding parameters (**TP**). Consequently, the MP can be defined as: **Media Parameter (MP)** is as a vector, $MP = \langle T, S, TP \rangle$.

Thus, based upon the **LO**, **HP**, **UP**, and **MP** definitions, we can define a **Content Adaptation Rule**, called **CAR**, to represent a *processed user request transaction* in our Adaptive Content Delivery Mechanism (ACDM).

Definition 4: Content Adaptation Rule (CAR),

$CAR = (LO, (B, HP, UP), MP_{set})$, where

- **B**: denotes the Bandwidth at the request time.
- $MP_{set} = \{MP_1, MP_2, \dots, MP_k\}$: denotes all associated MPs used to transcode all

physical media resources in a page.

For example, given a user request transaction, $\mathbf{CAR} 1 = ((LO_i, P_j), (B, HP, UP), MP_{set}) = (80, \langle 3, 400, 128, 480, 640, 16, 16, 44, S, WL \rangle, \langle 5, 7, CQD, JGBP, CSB, 0, 1, 0, 0, 0 \rangle, MP_{set})$, which denotes that a user uses a Pocket PC (3) with 400 Mhz, 128 MB, 480×640 resolution, 16 color depth under 80 kbps bandwidth (B) to retrieve the page (p_j) of LO_i , then the ACDM uses the MPs in MP_{set} , which are selected according to the B, HP, and UP, to transcode the physical resources in the P_j . Table 3.3 shows 14 CARs for the LO_{ij} .



Table 3.3: An Example of CAR for the same page (p_j) in a Learning Object (LO_i)

ID	Bandwidth (B)	Hardware Profile (HP)	User Preference (UP)
1	80	<3,400,128,480,640,16,16,44,S,WL>	<5,7,CQD,JGBP,CSB,0,1,0,0,0>
2	120	<4,500,384,800,400,16,16,44,S,WL>	<5,9,CDQ,GPBJ,SBC,0,0,0,0,0>
3	525	<4,133,96,360,400,16,16,4,S,WL>	<10,10,DQC,BGPI,SBC,0,0,0,0,0>
4	200	<4,133,768,360,400,4,U,U,S,WL>	<20,10,DQC,BGPI,SBC,1,1,1,0,0>
5	150	<3,400,160,640,400,4,U,U,S,WL>	<12,15,CQD,GPBJ,SBC,1,0,0,0,0>
6	180	<4,500,160,800,400,16,U,U,S,WL>	<7,10,QDC,JGBP,BSC,1,1,0,0,1>
7	80	<3,400,128,480,640,16,16,44,S,WL>	<5,7,CQD,JGBP,CSB,SBC,1,0,0,0>
8	100	<0,500,160,400,800,16,U,U,S,WL>	<7,10,QDC,JGBP,BSC,1,1,0,0,1>
9	152	<1,100,160,400,800,16,16,8,S,WL>	<7,10,QDC,JGBP,BSC,1,1,0,0,1>
10	90	<1,100,160,400,800,16,16,8,S,WL>	<14,5,DQC,JGBP,BSC,0,0,0,0,1>
11	75	<1,400,128,480,640,16,16,44,S,WL>	<5,7,CQD,JGBP,CSB,0,1,0,0,0>
12	50	<1,400,128,480,640,16,16,44,S,WL>	<5,7,CQD,JGBP,CSB,0,1,0,0,0>
13	70	<1,133,128,480,640,8,16,44,S,WL>	<5,7,CQD,JGBP,CSB,0,1,0,0,0>
14	72	<3,133,128,480,640,8,16,44,S,WL>	<5,7,CQD,JGBP,CSB,0,1,0,0,0>

Chapter 4

Content Adaptation Management Scheme (CAMS)

In this chapter, we will describe how to use the existing CARs to construct a **Content Adaptation Decision Tree**, called **CADT**, in CAMS. The CADT can be used to efficiently maintain, search, and retrieve the suitable adaptive content in LOR for users according to the user preference and then network bandwidth. As shown in Figure 4.1, The CAMS includes 3 processes to construct the CADT: **1) Clustering Process**, **2) Decision Tree Construction**, and **3) CADT Maintaining Process**, described in the following sections.

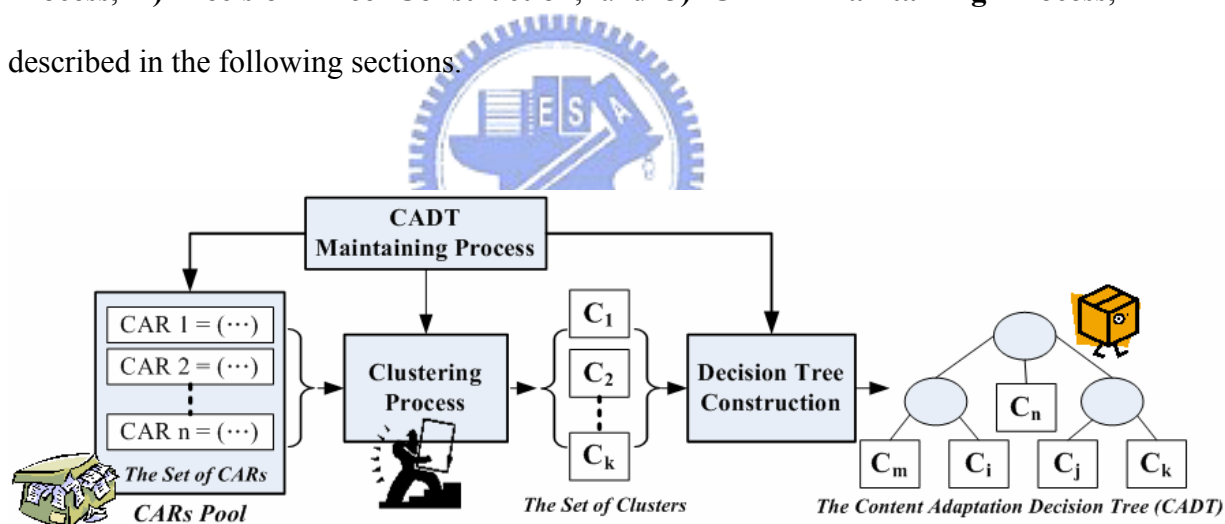


Figure 4.1: The Process of Content Adaptation Management Scheme (CAMS)

4.1 The Clustering Process of CAMS

As mentioned in Chapter 3, every user request with her/his preference like a transaction can be represented by a Content Adaptation Rule (CAR). As shown in Figure 4.1, in CAMS, all new CARs users requested are stored in a CARs Pool.

Therefore, we can apply distance based clustering algorithm to group these historical CARs into several clusters according to the user preferences (UP) of users, where every user in the same cluster has the similar user preference. However, it is difficult to determine the number of clusters for applying clustering approach. A clustering algorithm, called ISODATA [Hall&Ball], can dynamically change the number of clusters by lumping and splitting procedures and iteratively change the number of clusters for better result. Therefore, in this thesis, we apply the ISODATA clustering approach to group CARs into different clusters.

4.1.1 The Similarity Measure of Clustering Process

In order to apply the ISODATA clustering approach, we propose a similarity measure to efficiently estimate the similarity value between two CARs based on the UP. Because the attribute of a UP consists of **numerical attribute** and **symbolic attribute**, the similarity measure of UP can be formulized as follows:

Given two $UP_i = \langle a_1, a_2, \dots, a_n \rangle$ and $UP_j = \langle b_1, b_2, \dots, b_n \rangle$, the similarity measure of numerical attribute can be formulized as follows:

$$SimofNum_k = 1 - \frac{|a_k - b_k|}{Max - Min}$$

, where $1 \leq k \leq n$, the *Max* and the *Min* are the predefined maximum and minimum values of k_{th} attribute in a UP, respectively, described in the Definition 3.

Regarding the symbolic attribute in UP, such as the *Picture Property Ordering* (PPO), it has the value, *CSB*, which is like a *string*. Therefore, in order to compute the similarity between two symbolic attributes, their string based values can thus be divided into several pairs. For example, the string *CSB* can be divided into 3 pairs: $\{CS, CB, SB\}$. Accordingly, the number of the same pairs in two strings can be used to estimate the similarity. Thus, the similarity measure of two symbolic attributes can

be formulized as follows:

$$SimofSym_k = \frac{\text{The Number of TheSamePairs}(a_k, b_k)}{C_2^n}$$

, where $1 \leq k \leq n$, the C_2^n , the **Binominal Coefficient**, denotes the maximum number of pairs and the function *TheSamePairs()* indicates the same pair between two strings.

In addition, because different attributes in a UP may have different important degrees, we define a **Weight Vector (WV)** to adjust the important degree of every attribute in UP. Therefore, the Similarity Measure between two Ups can be formulated as follows:

Similarity_{UP}(UP_i, UP_j) = $\sum ((SimofNum_k(a_k, b_k) | (SimofSym_k(a_k, b_k)) \times w_k)$, where the $w_k \in \mathbf{WV}$ and $1 \leq k \leq n$.

Example 1:

Given two user preferences (UP), $UP_1 = \langle 5, 1/10, \text{DQC}, \text{JPGB}, \text{CSB}, 1, 0, 0, 1, 1 \rangle$ and $UP_2 = \langle 10, 1/7, \text{CQD}, \text{PJGB}, \text{SCB}, 1, 0, 1, 1, 1 \rangle$, and a predefined related attribute Weight Vector $\mathbf{WV} = \langle 0.35, 0.15, 0.05, 0.05, 0.05, 0.1, 0.05, 0.05, 0.05, 0.1 \rangle$.

We can thus apply the above similarity measure to compute the similarity between UP₁ and UP₂. For example, the similarity of the numerical attribute, *Delivery Time*

(DT), between UP₁ and UP₂ is $SimofNum_1 = 1 - \frac{|a_1 - b_1|}{Max - Min} = 1 - \frac{|5 - 10|}{30 - 0} = 1 - \frac{5}{30} = 0.83$.

Moreover, the similarity of the symbolic attribute, *Picture Format Ordering* (PFO), is

$$SimofSym_4 = \frac{\text{The Number of TheSamePairs}(\text{JPGB}, \text{PJGB})}{C_2^4} = \frac{\{\text{JG}, \text{JB}, \text{PG}, \text{PB}, \text{GB}\}}{C_2^4} = \frac{5}{6} =$$

0.83. Therefore, by the same way, the similarity between UP₁ and UP₂ is the

$$SimofSym_{UP}(UP_1, UP_2) = 0.83 \times 0.35 + 0.9 \times 0.15 + 0 \times 0.05 + 0.83 \times 0.05 + 0.66 \times 0.05 + 1 \times 0.1 + 1 \times 0.05 + 0 \times 0.05 + 1 \times 0.05 + 1 \times 0.1 = \mathbf{0.8}$$

4.1.2 The Clustering Algorithm based on ISODATA

In this thesis, the Delivery Time (DT) is the most important attribute in UP, so we use the DT value to estimate the centers of initial clusters of ISODATA. Accordingly, a **UP Clustering Algorithm (UPCALgo)** based on ISODATA is proposed to group these Ups into several clusters according to the Similarity Measure, shown in Algorithm 4.1. The Table 4.1 shows that after applying the **UPCALgo**, these CARs in Table 3.3 can be grouped into 4 clusters.



Algorithm 4.1: UP Clustering Algorithm (UPCALgo)

Symbols Definition:

DT: the Delivery Time (DT) in a user preference vectors (UP).

Upset: the set of UP.

K: the initial number of clusters.

T: the initial similarity threshold for selecting initial clusters.

C: a cluster with several user preference vectors (UP).

CC: the Center of Cluster.

C_{set}: the set of clusters I with the Center of Cluster (CC)

T_s: the split threshold (Standard Deviation) for splitting a cluster into 2 ones.

T_m: the merge threshold (Mean Distance) for merging 2 clusters into one.

T_n: the minimum number of the members in a Cluster for deleting a cluster.

Input: UP_{set}, K, T_s, T_m, T_n.

Output: The set of Clusters, C_{set}.

Step 1: Initial Clusters Selection:

Step 1.1: For I = 1 to K.

$$(1) T_i = \text{Min}(\text{DT}) + K \times \frac{\text{Max}(\text{DT}) - \text{Min}(\text{DT})}{(K - 1)}$$

(2) For each UP_i ∈ UP_{set},

if the |T_i - DT_i| is the minimum then insert UP_i into C_i with CC_i=UP_i
and then insert C_i into C_{set}.

Step 2: ISODATA Clustering Process:

Step 2.1: Execute the following sub-Steps (2.2-2.6) repeatedly until there is no difference between two iterations.

Step 2.2: Insert each UP_j ∈ UP_{set} into appropriate cluster C_i ∈ C_{set} according to the Similarity_{up}(CC_i, UP_j).

Step 2.3: Delete the C_i if I number of UP is less than T_n.

Step 2.4: Split a C_i into 2 clusters according to the T_s and T_n.

Step 2.5: Merge 2 clusters into one according to the T_m.

Step 2.6: Re-compute the Cluster Center (CC_i) for each C_i ∈ C_{set}.

Step 3: Output the C_{set}.

Table 4.1: The Result of Applying UP Clustering Algorithm

Cluster Label	ID of CAR
1	{2, 7, 11, 12, 13, 14}
2	{1, 3, 10}
3	{5, 6, 8, 9}
4	{4}

4.2 Decision Tree Construction

After UP Clustering process, every created cluster will be tagged with a cluster label as shown in Table 4.1. However, how to determine a suitable cluster for a new user request is an issue, and can be solved by the decision tree approach. Thus, based upon the Hardware Profiles (HP) in these CARs with cluster labels in Table 4.1, we can apply decision tree induction algorithm, ID3 [JRQUIN], to create a decision tree, called **Content Adaptation Decision Tree (CADT)**. Nevertheless, ID3 is able to process the symbolic value of attribute only, so the numerical attribute value of HP in Table 3.1, e.g., CPU, SM, etc., can be discretized by the following approach.

*In all HPs, λ and μ are the minimal and maximal values of attribute, respectively. Let $\Delta = (\lambda - \mu) / N$, where N is the number of desired discrete ranges. Then, a numeric value of attribute can be mapped into the symbolic value. For example, given N is 3, the corresponding symbolic values are **L** in $[\lambda, \lambda + \Delta]$, **M** in $[\lambda + \Delta, \lambda + 2\Delta]$, and **H** in $[\lambda + 2\Delta, \lambda + 3\Delta]$.*

Therefore, the numerical attribute of HP in Table 3.1 can be mapped into several discrete ranges, shown in Table 4.2. Afterward, for the data given in Table 3.3, 4.1,

and 4.2, the result of applying ID3 algorithm is shown in Figure 4.2.

Table 4.2: The Result of Mapping the Numerical Value in HP

Numerical Attribute	Representative Symbol
CPU Speed (CPU)	L: Low, M: Medium, H: High
System Memory (SM)	L: Low, LM: Low-Medium, MH: Medium-High, H: High,
Screen Horizontal Size (SHS)	T: Tiny, S: Small, M: Medium, L: Large
Screen Vertical Size (SVS)	T: Tiny, S: Small, M: Medium, L: Large

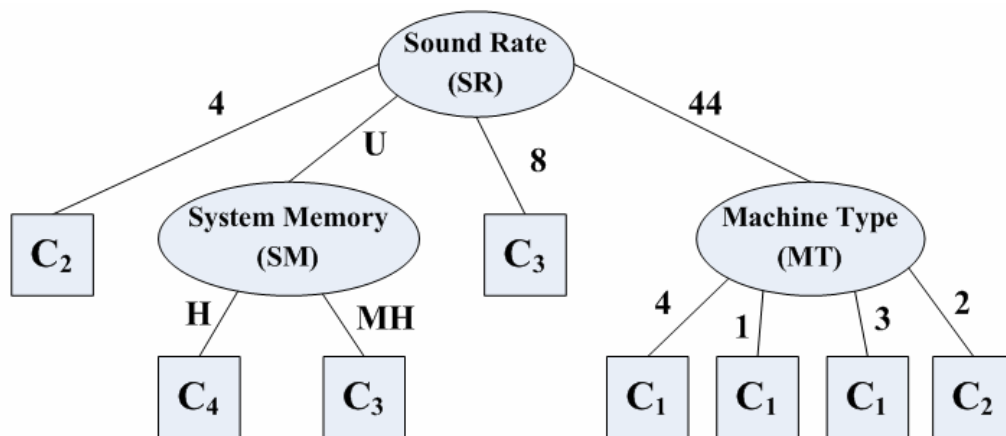


Figure 4.2: The Content Adaptation Decision Tree (CADT) based upon the Hardware Profiles (HP) in Table 3.3

4.3 CADT Maintain Process

As stated previously, after the Clustering Process and Decision Tree Construction in CAMS, all CARs in CARs Pool, which is a temporary buffer, can be grouped into several clusters and retrieved by CADT structure. However, how to assign a new CAR into a suitable cluster is an issue. Therefore, as shown in Figure 4.1, in CAMS, all new CARs users requested are stored in a CARs Pool first. While the number of

CARs in CAR Pool is more than a threshold, CAMS will automatically re-build the CADT by the Clustering and Decision Tree process. Then, these processed CARs in CARs Pool will be shifted to final storage.



Chapter 5

Adaptive Content Delivering Process

5.1 Content Adaptation Process

In ACDM, for a new User Request (UR), if there is no suitable existing adaptive content to be delivered, the **Content Adaptation Process (CAP)** will automatically determine an appropriate MP_{set} described in Chapter 3.2 to transcoding all media resources in a desired page according to the requirement in UR. Thus, the UP with corresponding Mpset can represent a processed user request transaction. The process is described as follows.

Given a new User Request, $UR = (LO_{ij}, (B, HP, UP))$, where $UP = \langle DT, Ratio, PPO, PFO, APO, FPS, BPS, IS, BS, AS \rangle$ and the associated media resources and corresponding size in the page p_j are $R_j = \{r_1, r_2, \dots, r_k\}$ and $RS_j = \{rs_1, rs_2, \dots, rs_k\}$, respectively. Furthermore, we define the resultant media resources of p_j , call **Scaled Resource Size (SRS)**, $SRS_j = \{srs_1, srs_2, \dots, srs_k\}$, which can be used to select the appropriate MP_{set} from MP database.

Based on UR, we can estimate the maximum file size we can use, called $MaxFS_{Esti} = B \times DT$.

Moreover, we have to compute the maximum file size of original media resources in page p_j , as follows:

$MaxFS_{orig} = (MaxImageSize \times (1 - Ratio)) + (BackAudioSize \times Ratio)$, where

$$MaxImageSize = FPS \times \sum rs + BPS \times \sum rs + IS \times \sum rs + BS \times rs,$$

where the $rs \in \mathbf{RS}_j$ is the *Foreground, Bar, Icon, or Background*, depending on the picture type. Afterward, we can get the **Scale Weight (SW)** = $\mathbf{MaxFS}_{\text{Esti}} / \mathbf{MaxFS}_{\text{orig}}$, which can be used to scale each media resource size into suitable one. Finally, the \mathbf{SRS}_j , can be obtained by:

$$\mathbf{srs}_\gamma = \mathbf{rs}_\gamma \times \mathbf{SW}, \text{ where } \mathbf{srs}_\gamma \in \mathbf{SRS}_j, \mathbf{rs}_\gamma \in \mathbf{RS}_j, 1 \leq \gamma \leq k, .$$

Therefore, based on the above equations, we propose a Content Adaptation Process (CAP) Algorithm (CAPALgo) to select an appropriate \mathbf{MP}_{set} to transcode content into suitable version, described in Algorithm 5.1.



Algorithm 5.1: Content Adaptation Process (CAP) Algorithm (CAPALgo)

Symbols Definition:

MP_{db} : denote the MP database storing the preprocessed MP.

MP_{temp} :store several temporary MP.

$R_j = \{r_1, r_2, \dots, r_k\}$: denote the media resources in page p_j .

$SRS_j = \{srs_1, srs_2, \dots, srs_k\}$:denote the **Scaled Resource Size (SRS)** Set for page p_j .

Input: a new user request, **UR**.

Output: a new **CAR** consisting of **UR** and an MP_{set} .

Step 1: for each media resource with switch attribute is true, $r_\gamma \in R_j$, in a p_j .

Step 1.1: if the media type of r_γ is a *Picture*.

Then execute the following sub-steps

(1) insert all $MP_i \in MP_{db}$ into MP_{temp} according to attribute $PFO \in UP$

(2) delete the $MP_i \in MP_{temp}$ if its *horizontal resolution* $> (SHS \in HP)$
and its size $(S) > srs_\gamma \in SRS_j$.

(3) keep each $MP_i \in MP_{temp}$ with higher value according to the selecting
priority order of attribute $PFO \in UP$, as shown in Table 3.2.

(4) select the $MP_i \in MP_{temp}$ with the highest size (S) and then insert into

MP_{set} .

Else if the media type of r_γ is an *Audio*.

(1) insert all $MP_i \in MP_{db}$ into MP_{temp} if its its size $(S) \leq srs_\gamma \in SRS_j$.

(3) keep each $MP_i \in MP_{temp}$ with higher value according to the selecting
priority order of attribute $APO \in UP$, as shown in Table 3.2

(4) select the $MP_i \in MP_{temp}$ with the highest size (S) and then insert into

MP_{set} .

Step 2: output a new **CAR** consisting of **UR** and an MP_{set} .

Example :

For example, given a user request transaction, $CAR 1 = ((LO_i, P_j), (B, HP, UP), MP_{set}) = (80, \langle 3, 400, 128, 480, 640, 16, 16, 44, S, WL \rangle, \langle 5, 7, CQD, JGBP, CSB, 0, 1, 0, 0, 0 \rangle, MP_{set})$.

5.2 Adaptation Decision Process (ADP)

In CAMS, the CADT can be used to efficiently search, retrieve, and maintain the historical CARs users requested previously. Therefore, the desired adaptive contents can be fast delivered to users if there is a similar existing user request held by CADT. However, how to efficiently deliver an appropriate adaptive content from the existing CARs or redo the aforementioned Content Adaptation Process is our concern. Accordingly, we propose an **Adaptation Decision Process Algorithm** (ADPAlgo) to efficiently process the adaptive content decision. The ADPAlgo is shown in Algorithm 5.2 and an example is described as follows.

Example 2:

Given a new User Request (UR), $UR = (LO_{ij}, (B, HP, UP)) = (LO, (90 \text{ KB}, <3,400,512,480,640,16,16,44,S,WL>, <5,1/7,CQD,JGBP,CSB,1,1,1,1,1>))$ and a new CAR 15= $(LO_{ik}, (150 \text{ KB}, <1, 133, 128, 480, 640, 16, 16, 44, S, WL>, <12, 1/7, QCD, GJBP, CSB, 1, 0, 1, 1, 1>))$ in **CARs Pool**, according to the CADT and Adaptation Decision Process Algorithm, we can find the rule: *if Sound Rate (SR) = '44' and Machine Type (MT) = '3' then 'C₁'*, so that we can use the CAR ID, {2, 7, 11, 12, 13, 14}, of C₁ in Table 4.1 and CAR 15 in **CARs Pool** to select a suitable CAR (**Step 1-Step 2**). Then, the CAR 15 is deleted ($LO_{ik} \neq LO_{ij}$) and CAR 2, 12, and 13 are deleted while α is 0.8 (**Step 3-Step 4**). Afterward, the CAR 7 with 9 similar attributes and 0.95 similarity value compared with UR is a suitable CAR for user (**Step 5-Step 6**). However, because CAR 7 is not the same as UR fully, based on UR, a new CAR, 16, will be created by the Content Adaptation Process (CAP) and stored in CARs Pool (**Step 8**). Thus, the CARs Pool will hold two new CARs, i.e., {15, 16} and the

adaptive contents based on CAR 7 will be delivered to a user directly without transcoding latency.



Algorithm 5.2: Adaptation Decision Process Algorithm (ADPAlgo)

Symbol Definition:

CAR_{set}: store several historical CARs

UR: denote a user request, i.e., $UR=(LO, (B, HP, UP))$.

CAR_{new}: store the new CAR created according to the **UR**.

A: denote the acceptable percent threshold of bandwidth deviation.

B: denote the minimum number of the similar attributes of HP between **CAR_{set}** and **UR**.

γ : denote the acceptable threshold of bandwidth deviation.

Input: a **UR**

Output: a suitable **CAR**

Step 1: If the **CADT** is not Empty,

Then use the HP in **UR** to traverse the **CADT** for finding the suitable cluster with similar HP.

Step 2: Insert CARs into **CAR_{set}** from the selected Cluster in **CADT** and **CARs Pool**.

Step 3: Delete these CARs from **CAR_{set}**, if $(LO_{ij} \in \text{CAR}_{set}) \neq (LO_{mn} \in \text{UR})$.

Step 4: Delete these CARs from **CAR_{set}**, if $|(B \in \text{CAR}_{set}) - (B \in \text{UR})| \geq \alpha \times B \in \text{UR}$.

Step 5: Delete these CARs from **CAR_{set}**, if the number of HP attributes with similar value in CAR compared with **UR** $< \beta$.

Step 6: Delete these CARs from **CAR_{set}**, if the similarity between CAR in **CAR_{set}** and **UR** according to the $\text{Similarity}_{up}() < \gamma$.

Step 7: If \exists a $\text{CAR} \in \text{CAR}_{set}$ whose attribute values in HP and UP is the same as **UR**,

Then goto **Step 9**.

Step 8: do the Content Adaptation Process (CAP) according to the UP in **UR** and create the **CAR_{new}** stored in **CARs Pool**.

Step 9: If **CAR_{set}** is not empty,

Then Output the CAR with the highest similarity in **CAR_{set}**.

Else Output the **CAR_{new}**.

5.3 Content Synthesizer

As stated previously, regarding a new **UR** without suitable existing adaptive content to be delivered, the Content Adaptation Process (CAP) will decide a corresponding \mathbf{MP}_{set} to transcode the associated media resources. However, some media resource may not be shown on user's handheld device depending on the associated switch attributes, e.g., FPS, BPS, IS, BS, and AS in UP. Therefore, given a page has n media resources and its corresponding $\mathbf{MP}_{\text{set}} = \{\mathbf{MP}_1, \mathbf{MP}_2, \dots, \mathbf{MP}_m\}$, where $1 \leq m \leq n$, which implies that the $(n-m)$ resources need not be transcoded and shown. Thus, in order to hint users, these unshown media resources will be replaced by some additional annotations. Moreover, for efficiently manipulating the diverse versions of content, the original HTML in a page will be transformed into an XHTML[XHTML] due to its well form and tree structure like DOM [DOM].



Algorithm 5.3: Adapted Content Generation

Symbol Definition:

r_γ : denote the original media resource in a page.

Tr_γ : denote the transcoded media resource.

Input: a UR with corresponding MP_{set} and LO_{ij} .

Output: a transcoding content version, XHTML.

Step 1: for each media resource, r_γ , in a page p_i .

Step 1.1: apply $MP_\gamma \in MP_{set}$, to transcode the r_γ into the tr_γ .

Step 2: transform the original HTML into XHTML format

Step 3: replace all r_γ by tr_γ into the XHTML.

Step 4: replace all unshown media resources by useful annotation.

Step 5: output the XHTML with associated transcoded media resources.



Chapter 6

Implementation and Experimental Results

The ACDM prototypical system is developed based on Apache Server and PHP, Perl and C Language. Except screenshots of current implementation, the experimental results are good.

6.1 The System of ACDM

The ACDM Web Server has built up with SCORM-based learning object repository to provide mobile devices viewing learning object online. Figure 6.1 shows the main flow of ACDM Web Server.

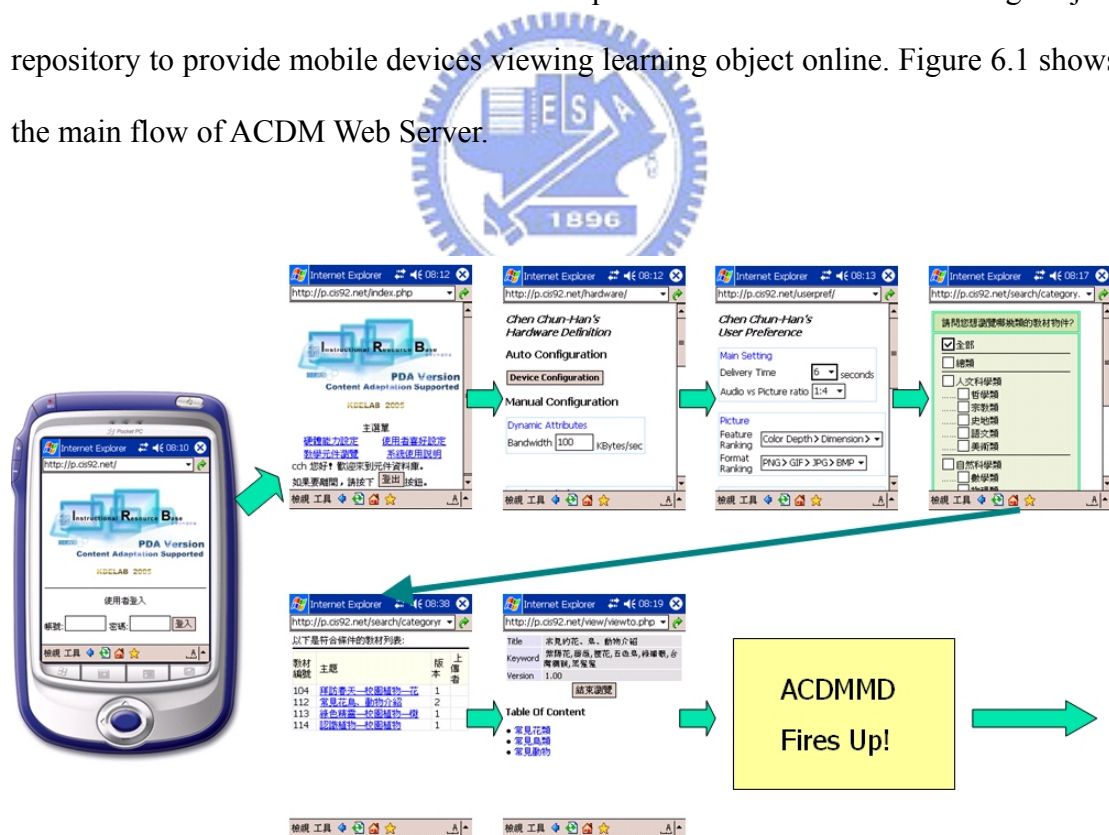


Figure 6.1: Main flow for a user to retrieve a learning object

Figure 6.2 shows the content adaptation results comparison when meeting the

different bandwidth setting. The user also specifies the picture format's ordering (PFO). ACDM Web Server will do content adaptation according to the preferences and suppress the sizes of the media objects to meet the delivery time condition.

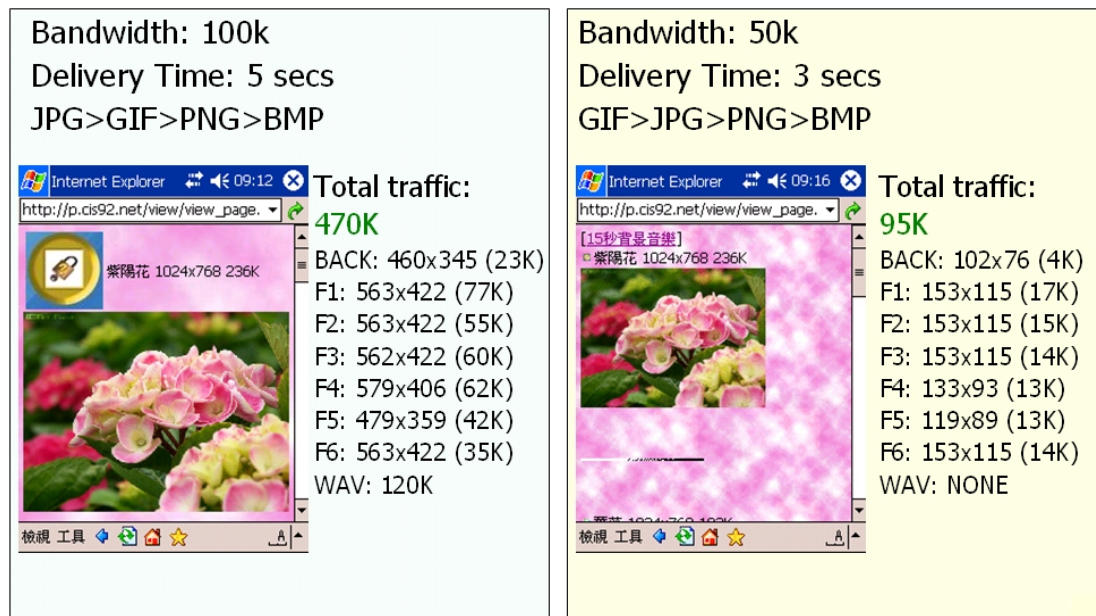


Figure 6.2: Content Adaptation

Figure 6.3 shows the adaptation result can be generated in accordance with the switch settings of user preference. If the user switches off the WAV file, BACKGROUND, ICON, and BAR pictures, they would not be displayed. There are two influences of the result. First, the disabled media object will be substituted with the annotation one and be transformed to a hyperlink. Second, the redundant traffic size will be split into other media objects to improve their display qualities. For example, the F1 picture has been upgraded from 77Kbytes to 112Kbytes.

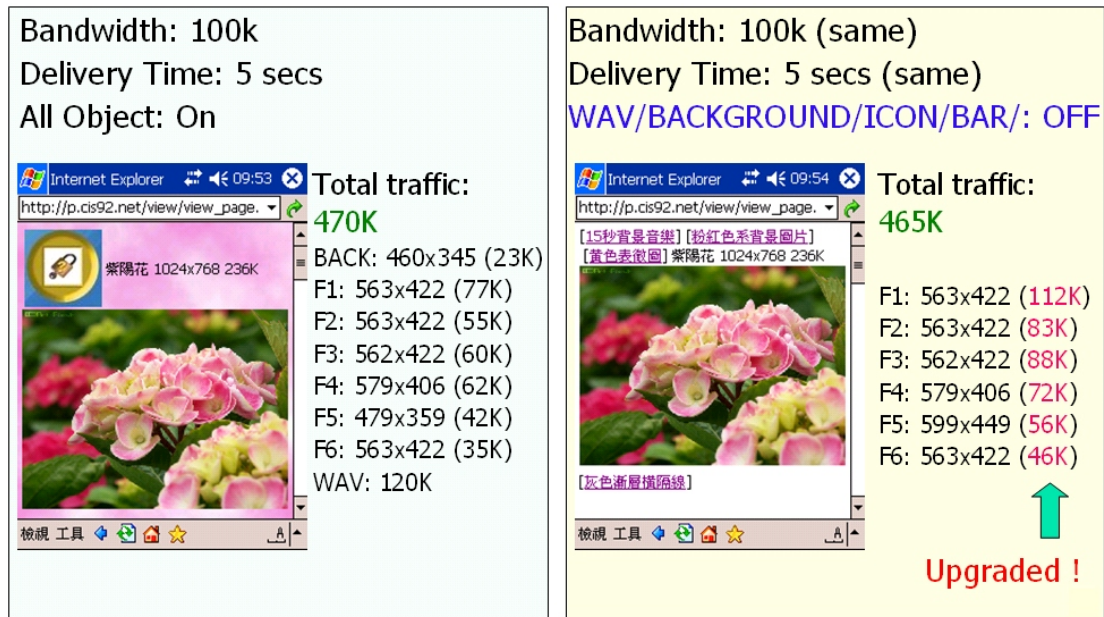


Figure 6.3: Preference-based Transformation

Figure 6.4 shows the effect of “**progressive adaptation**”. We assume there is a previous retrieval that the user tries to get the learning object with turning off the background. When next user with the same user preference except for turning on the background tries to get the same learning object, the adaptation result is still the same with previous one (no background). It is because the similarities between two preferences are high enough to adopt the previous adaptation rule. At the same time, ADP invokes CAMS and Content Synthesizer to do the real transformation with the true condition which the user has. Therefore, when the other learner tries to retrieve the same learning object again, the background will be shown.

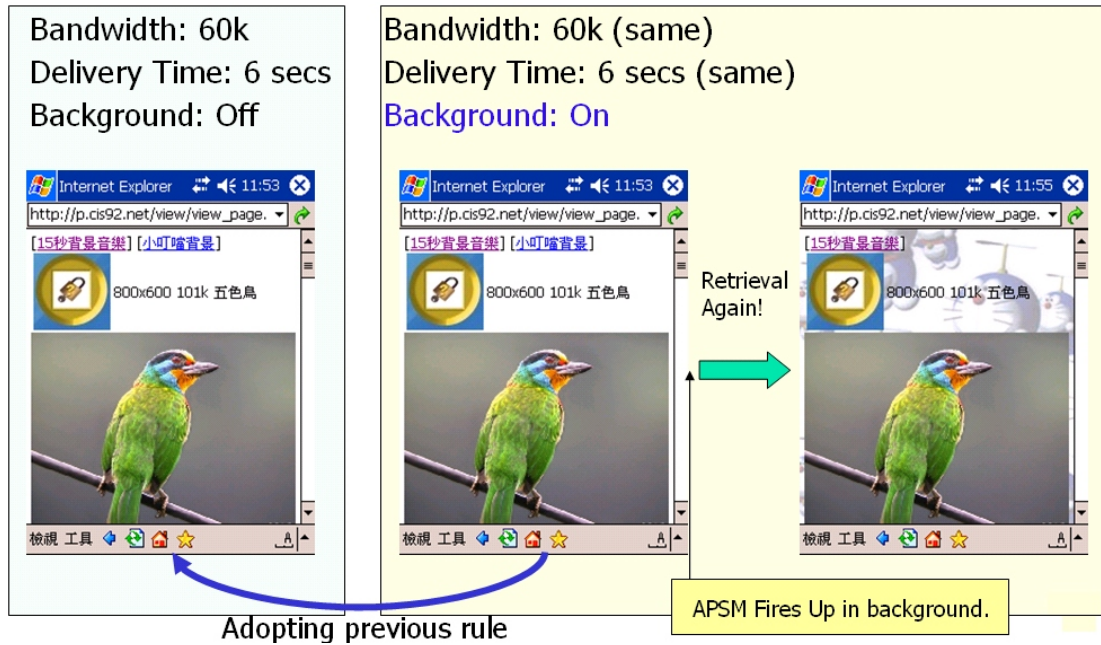


Figure 6.4: Progressive Adaptation

The final screenshot is the administrative interface for the administrator to maintain or observe the ACDM Web Server conditions. It provides two buttons to trigger the task of model construction. The first button is “Assign cluster label”. It first runs UPCALgo to get the new rule clusters and assign the cluster label to each rule. The second button is “Rebuild decision tree”. It triggers CAMS to redo classification with hardware profile and saves the result into memory. The rest of the components in Figure 6.5 show the adaptation rules, decision tree, and the detail of rule clusters. Figure 6.5 shows the ACMD Web Server Monitoring.

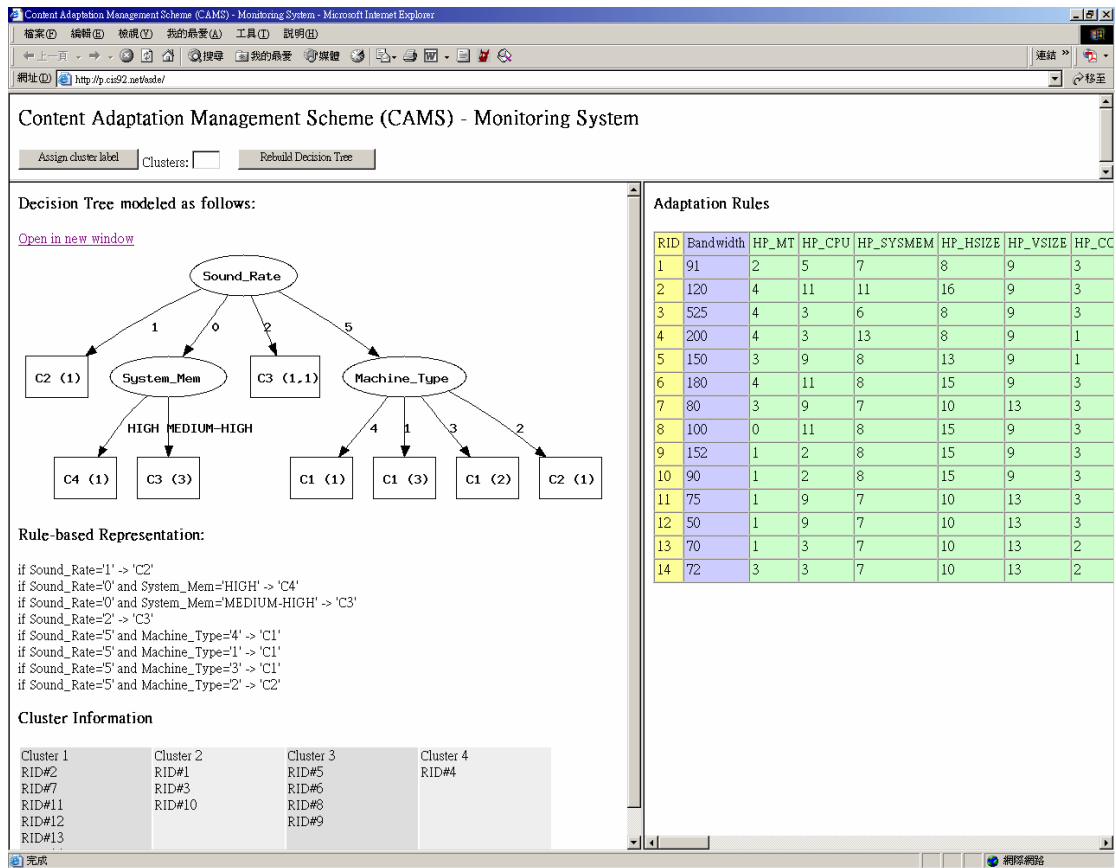


Figure 6.5: ACDM Web Server Monitoring

6.2 Experiments

Several experiments have been conducted to prove the ACDM system well. Assume there is a learning object which contains a WAV file with 660,768 bytes and six pictures with 1,016,392 bytes. The original size is about 1.8 Mbytes. The user specified the maximum tolerable delivery time to 5 seconds. We observe the transmission result with the various bandwidth settings.

In the first experiment, we assume there are no prior connections existing before. When the first user retrieves the learning object, it could take about 3 seconds to do the content adaptation (by Content Synthesizer), and the later connections, the content could be transmit immediately; hence the transmission time is in the control of 5

seconds. This experiment also shows ACDM always keeps the delivery time controlled with various bandwidth settings. Figure 6.7 shows the result. Figure 6.8 shows the total transmission size variation.

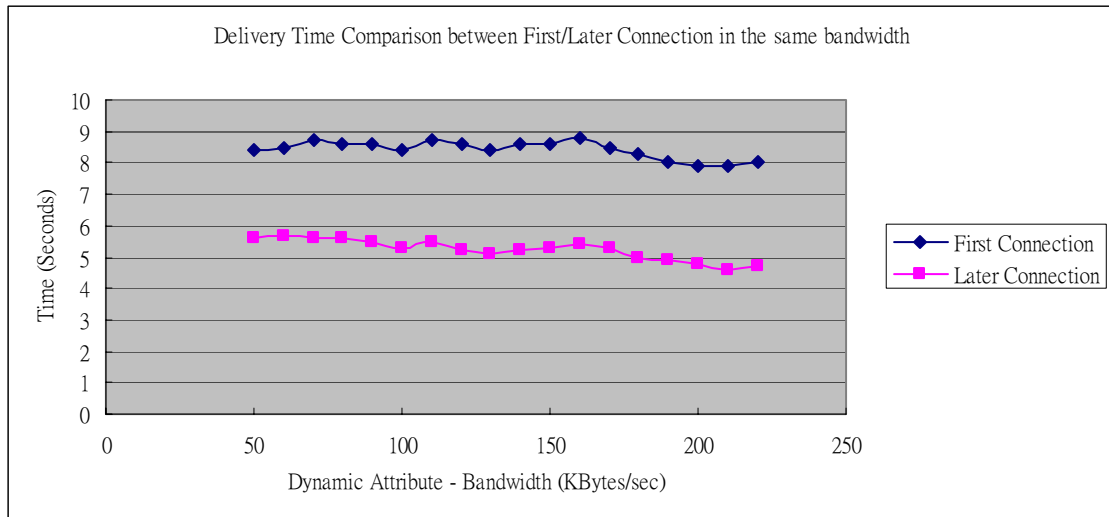


Figure 6.7: Exp 1 – Delivery Time with various bandwidth settings

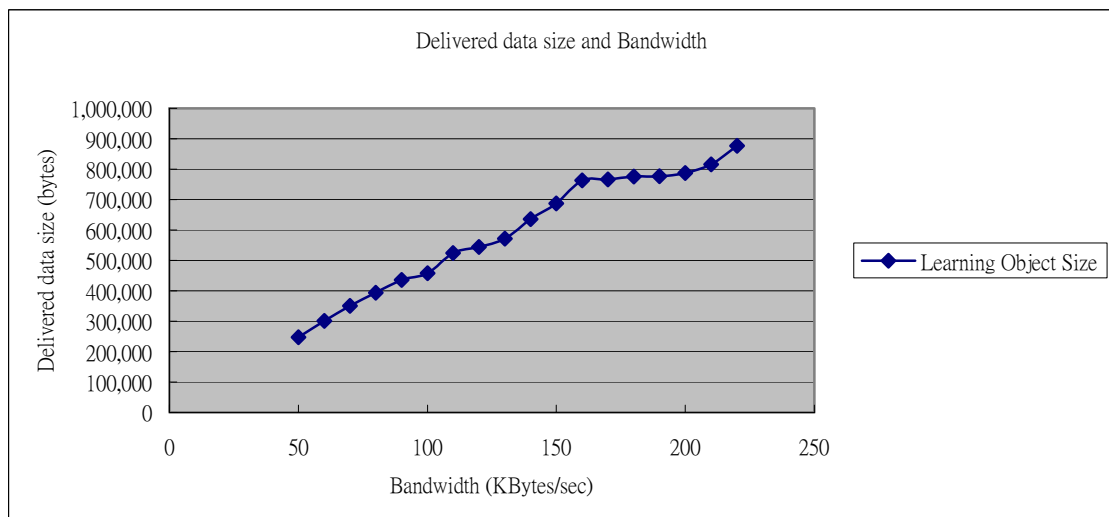


Figure 6.8: Exp 1-B – Observations of total transmission size

The second experiment is conducted to compare the transmission time between Traditional Web Server and ACDM-based Web Server. In virtue of the simplicity of the traditional web server, since it sends whole content without any adaptations, the transmission time is long when the bandwidth is low. ACDM can overcome this suffering.

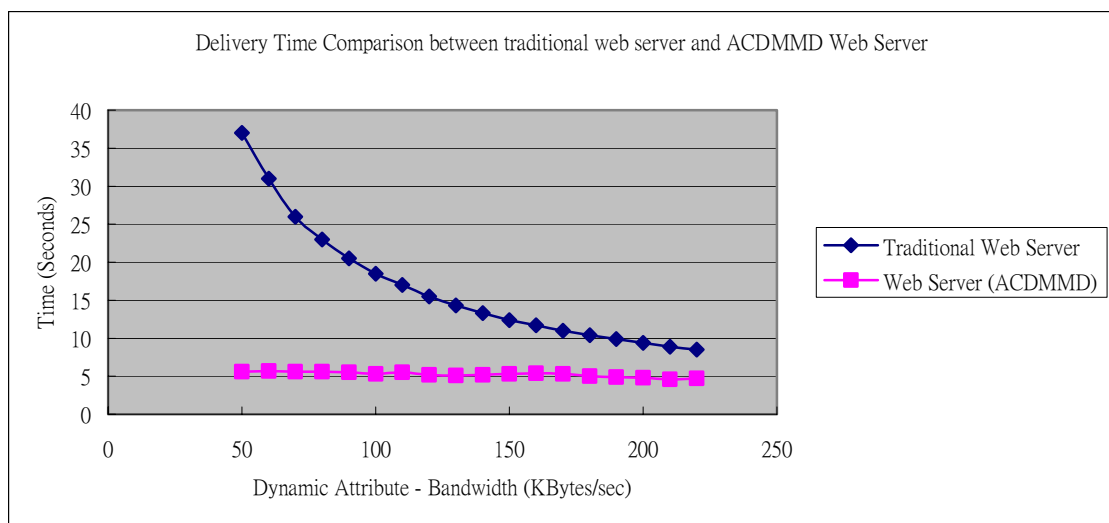


Figure 6.9: Exp 2 – Comparison with traditional web server

The third experiment compares Annotation-based Web Server and ACDM-based Web Server. In Annotation-based Web Server, there are several content versions of learning object defined in the authoring stage. Hence it may have various versions of content to deliver. For example, there are three existing content versions and they are suitable of bandwidth settings with 140Kbytes/sec, 170Kbytes/sec and 200Kbytes/sec. The experiment result shows the Annotation-based Web Server suffers a long delivery time when the bandwidth is < 100k. It is because when the bandwidth is very low, sending the content version of 140Kbytes/sec causes a long delivery time.

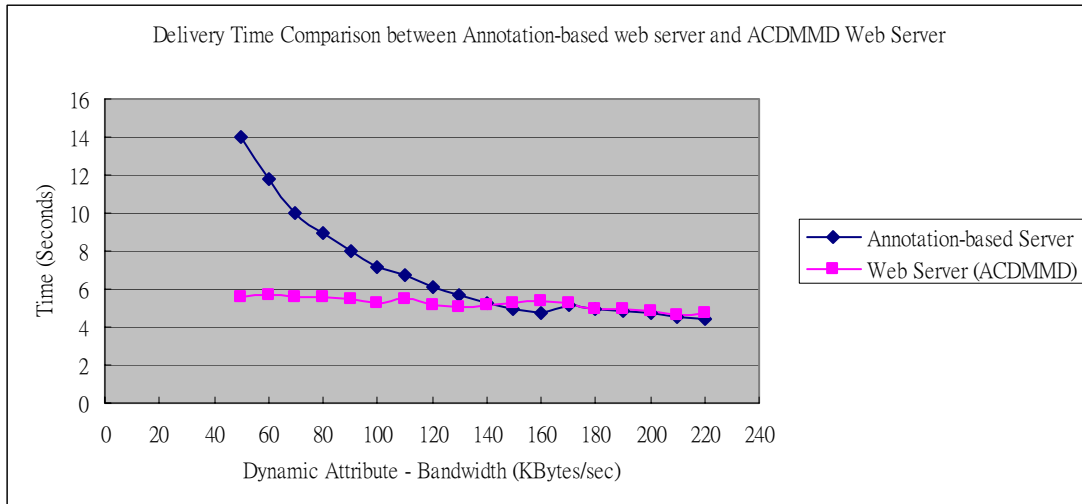


Figure 6.10: Exp 3 – Comparison with Annotation-based Server

The fourth experiment shows that ACDMMD Web Server has the capability to do bandwidth estimation dynamically. It overcomes the problem when network characteristic is changed. The first connection may suffer long delivery time because the bandwidth user specified differs from the actual condition. After the transmission is done, the bandwidth value is re-calculated automatically. The later connection may regain the delivery time in control. The following experiment assumes the user specified the maximum tolerable delivery time to 5 seconds.

Table 6.1: Dynamic bandwidth re-estimation

Bandwidth (User)	Bandwidth (Actually)	Transmission Time	Remark
200Kb	200K	5.3 seconds	
200Kb	150K	7 seconds	*
150Kb (Auto Update)	150K	5.2 seconds	
150Kb	100K	7.8 seconds	*
100Kb (Auto Update)	100	5.4 seconds	
100Kb	50K	10.5 seconds	*
50Kb (Auto Update)	50K	5.2 seconds	

● : Time excessive.

Chapter 7

Conclusions

In this paper, we propose an **Adaptive Content Delivery Mechanism**, called **ACDM**, which can efficiently manage a large number of historical user requests, and intelligently deliver a proper adaptive content with higher fidelity from LOR to users directly and then prepare a transcoded content version for next similar request. The **ACDM** includes **Adaptation Data Format Definition Phase** and **Adaptive Content Delivery Phase**. The former defines an adaptation data format, called **Content Adaptation Rule (CAR)**, based upon CC/PP, UAProf, etc. In order to efficiently deliver the suitable content with associated learning resources to users in accordance with their user preferences, hardware capabilities, and variable wireless bandwidth, the latter consists of **1) Content Adaptation Management Scheme (CAMS)**: applies clustering approach and decision tree approach successively to create a **Content Adaptation Decision Tree (CADT)**, which can be used to predict the appropriate adaptive contents from the LOR, **2) Adaptation Decision Process (ADP)**: proposes an **Adaptation Decision Process Algorithm** to intelligently determine a suitable version of the existing adaptive content based on the **CADT**, and **3) Content Synthesizer**: transcodes the content if necessary. For evaluating our proposed approach, an ACDM prototypical system is developed. Furthermore, the experimental results show that the ACDM is workable and beneficial. In the near future, we will also deploy the ACDM system on the general web server, not only the SCORM learning object repository. Besides, the management scheme will be enhanced to efficiently maintain the huge number of adapted content versions in the ACDM storage.

Bibliography

- [AICC] AICC (Aviation Industry CBT Consortium), <http://www.aicc.org>
- [ARIADNE] ARIADNE (Alliance for Remote Instructional and Authoring and Distribution Networks for Europe), <http://www.ariadne-eu.org>
- [CCPP] Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0, <http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115>
- [CWZ03] Chen, Y., Ma, W.Y., and Zhang, H.J. "Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices." Proc. WWW'03, Budapest, Hungary, May 2003
- [DOM] Document Object Model. <http://www.w3.org/DOM/>
- [FIPA02] Foundation for Intelligent Physical Agent, (FIPA), 2002, "Device Ontology Specification," SI00091E edition, Retrieve 2005 from http://www.fipa.org/specs/fipa00091/SI00091E.html#_Toc26671148
- [GF+02] Gonzalez-Castano, F., L. Anido-Rifon, et al. (2002). « A new transcoding technique for PDA browsers, based on content hierarchy. » HUMAN COMPUTER INTERACTION WITH MOBILE DEVICES 2411 : 69-80.
- [Hall&Ball] [Hall&Ball] D.J. Hall and G.B. Ball, "ISODATA : A Novel Method of Data Analysis and Pattern Classification", Technical report, Stanford Research Institute, Menlo park CA, 1965.
- [HFTS] Handy-Fit-To-Screen. (Microsoft Pocket PC) <http://www.microsoft.com/mobile/pocketpc/features/articles/web.asp>
- [HMZ04] Hinz, M., Z. Fiala, et al. (2004). "Personalization-based optimization of Web interfaces for mobile devices." MOBILE HUMAN-COMPUTER INTERACTION – MOBILEHCI 2004, PROCEEDINGS 3160: 204-215.

- [IMAGICK] ImageMagicK, <http://www.imagemagick.org/>
- [IMS] IMS (Instructional Management System), <http://www.imsproject.org>
- [JRQUIN] J.R. Quinlan. Induction of decision trees. Machine Learning, 1(1):81–106, 1986.
- [LTSC] LTSC (IEEE Learning Technology Standards Committee), <http://ltsc.ieee.org/wg12>
- [LWF03] Lum, W. and F. Lau (2003). “User-centric content negotiation for effective adaptation service in mobile computing.” IEEE TRANSACTIONS ON SOFTWARE ENGINEERING 29(12): 1100-1111.
- [OHA01] O. Buyukkokten, H. Garcia-Molina and A. Paepcke, “Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices”. Proc. Of WWW-10, 2001, Hong kong
- [OOH02] O. Buyukkokten, Oliver Kaljuvee, Hector Garcia-Molina, Andreas Paepcke, and Terry Winograd. “Efficient web browsing on handheld devices using page and form summarization.” ACM Transactions on Information Systems (TOIS), 20(1) :82–115, 2002.
- [PWB3C] PalmSource Web Browser 3.0 (3COM Palm PDA)
http://www.palmsource.com/palms/Web_Browser.pdf
- [RFC2616-14] Hypertext Transfer Protocol -- HTTP/1.1, Header Field Definitions
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- [SCORM] Sharable Content Object Reference Model (SCORM), <http://www.adlnet.org>
- [SOX] SoX. <http://sox.sourceforge.net/>
- [TNS03] T. Lemlouma and N. Layaida: Adapted Content Delivery for Different Contexts, SAINT 2003 Conf., Orlando, Florida, USA, January 27-31, 2003. IEEE, pp. 190-197.
- [UAPROF] User Agent Profiling Specification v1.1
<http://www.openmobilealliance.org/tech/affiliates/wap/>

wap-248-uaprof-20011020-a.pdf

- [VLC04] Villard, L., C. Roisin, et al. (2004). “An XML-based multimedia document processing model for content adaptation.” DIGITAL DOCUMENTS: SYSTEMS AND PRINCIPLES 2023: 104-119.
- [XHTML] XHTML. <http://www.w3.org/TR/xhtml1>
- [XSLT] W3C. XSL Transformations (XSLT) Version 1.0. W3C Recommendation. <http://www.w3.org/TR/1999/REC-xslt-19991116>.
- [XWW04] Xinyi Yin, Wee Sun Lee, “Using link analysis to improve layout on mobile devices”, Proc. WWW’04, New York, NY, USA.



REV 2.13