User Model User-Adap Inter (2011) 21:5–49
DOI 10.1007/s11257-010-9094-0

ORIGINAL PAPER

A personalized learning content adaptation mechanism to meet diverse user needs in mobile learning environments

**Jun-Ming Su · Shian-Shyong Tseng ·
Huan-Yu Lin · Chun-Han Chen**

Received: 15 April 2010 / Accepted in revised form: 18 December 2010 /
Published online: 28 January 2011
© Springer Science+Business Media B.V. 2011

Abstract With the heterogeneous proliferation of mobile devices, the delivery of learning materials on such devices becomes subject to more and more requirements. Personalized learning content adaptation, therefore, becomes increasingly important to meet the diverse needs imposed by devices, users, usage contexts, and infrastructure. Historical server logs offer a wealth of information on hardware capabilities, learners' preferences, and network conditions, which can be utilized to respond to a new user request with the personalized learning content created from a previous similar request. In this paper, we propose a Personalized Learning Content Adaptation Mechanism (PLCAM), which applies data mining techniques, including clustering and decision tree approaches, to efficiently manage a large number of historical learners' requests. The proposed method will intelligently and directly deliver proper personalized learning content with higher fidelity from the Sharable Content Object Reference Model (SCORM)-compliant Learning Object Repository (LOR) by means of the proposed adaptation decision and content synthesis processes. Furthermore, the experimental results indicate that it is efficient and is expected to prove beneficial to learners.

J.-M. Su (✉)
Department of Information and Learning Technology,
National University of Tainan, Tainan 700, Taiwan
e-mail: junming.su@gmail.com

S.-S. Tseng
Department of Applied Informatics and Multimedia, Asia University, Taichung 413, Taiwan
e-mail: sstseng@asia.edu.tw

H.-Y. Lin · C.-H. Chen
Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan
e-mail: linhuanyu@gmail.com

C.-H. Chen
e-mail: is88050@cis.nctu.edu.tw

Keywords Personalized learning content · Content adaptation · Mobile learning environment · Data mining · Learning object repository

1 Introduction

The proliferation of e-learning has been sparked by the rapid development of the Internet. This educational approach has become more and more popular because it allows learners to conveniently study anytime, anywhere; no longer are students restricted to the confines of a classroom. Due to the emergence of heterogeneous mobile devices and wireless technologies, such as cell/smartphones, Personal Digital Assistants (PDAs), netbooks, and other similar mechanisms, the requirements for delivering and displaying learning content on mobile devices are increasing rapidly (Tretiakov and Kinshuk 2004; Yang et al. 2007a; Chang et al. 2008; Queirós and Pinto 2009; Pettersson and Gil 2010). While the types of mobile devices and the available e-learning subject matter proliferate, a large gap still exists between traditional desktop and mobile-based systems. That is, most existing learning content is created for desktop devices, such as Personal Computers (PCs), and not for the mobile devices currently used. Consequently, the insufficient hardware capabilities, e.g., the limited computation power, memory, screen size, and unstable bandwidth of a wireless network, have led to poor navigation experience and unfavorable presentation of learning content. In many cases, it is difficult to display the original content, which is customized to a desktop computer and a larger monitor, on the display of a handheld device in a usable and efficient manner. How learning content is adapted to meet the needs of various users via various mobile devices needs to be addressed and improved.

This leads to the issue of *content adaptation*, which concerns the act of transforming learning content to adapt to any mobile device capabilities. How to successfully transform existing learning content into a suitable version that can be efficiently delivered and displayed to meet the needs of learners and of mobile devices has been a growing area of study. In order to solve this problem, a significant amount of research has been conducted that proposes numerous content adaptation approaches. Two notable approaches include: (1) static adaptation, which preprocesses and stores multiple versions of the content; and (2) dynamic adaptation, which adapts the content real-time during the user's request (Fudzee and Abawajy 2008).

The static adaptation approach is capable of reducing the time it takes for learning content to download, but it requires a preprocessing task and larger storage allocation (Mohan et al. 1999; Villard et al. 2000; Hinz et al. 2004). In contrast, the dynamic adaptation approach can apply content structure analysis (Buyukkokten et al. 2001, 2002; González-Castaño et al. 2002; Chen et al. 2003; Yin and Lee 2004; Ramaswamy et al. 2005; Laakko and Hiltunen 2005; Yang et al. 2007b; He et al. 2007; Kim et al. 2008; Hsiao et al. 2008) to dynamically transcode, rearrange the layout, and distillate the content to improve the delivery latency and meet the devices' capabilities. Alternatively, it can apply context-based adaptation (Lum and Lau 2003; Mohomed et al. 2004, 2006a,b, 2007; Lee et al. 2006; Yang et al. 2008) to consider both the environmental context and the preference of the requesting user to offer more precisely adapted content.

However, the content quality of mobile devices is limited to fixed page fragments, which are compatible with content adaptation processing approaches. Furthermore, the information of content, e.g., text and semantic relations, may be lost during the decomposition or summarization processes. Based on our review of the literature (Beekhoven et al. 2003, Chen and Mizoguchi 1999; Dewhurst et al. 2000; Chen et al. 2000; Smith 2001; McIlroy et al. 2001; Riding and Cheema 1991; Wilson 2000; Franzoni et al. 2008), different students have different learning approaches and preferences for learning. That is, not all students are the same, nor are their preferences for learning. Students, therefore, favor various ways to receive information and acquire knowledge, such as: content type preference (Gilbert and Han 1999; Stern and Woolf 2000), presentation style, cognitive styles (Riding and Cheema 1991), and learning style (Kolb 1976, 2004), among others.

In mobile learning environments, the response time of requested information can significantly impact and influence learning performance. With the proliferation of mobile device usage for educational purposes, a learner must be able to retrieve requested information with few hindrances. For example, although a specific request for learning content has been appropriately tailored for the learner, if this content is delivered too slowly, or if the overall quality is poor, it will explicitly affect the learner's satisfaction (Muntean 2008; Ding et al. 2010). Accordingly, a given learner's satisfaction will be affected not only by his or her perception and preference, but also by the network-related issues and mobile device factors (Muntean 2008). Most existing research and systems either separately or partially considered the features of mobile devices, network conditions, and learner preferences (Zhao et al. 2008; Nimmagadda et al. 2010). Therefore, a pressing issue is how to consider not only the mobile devices' capabilities, but also the available network conditions and learners' diverse preferences in order to efficiently provide properly personalized learning content in mobile learning environments (Tong et al. 2006; Mylonas et al. 2007; Basaeed et al. 2007; Zhao et al. 2008; Franzoni et al. 2008; Muntean 2008; Pettersson and Gil 2010; Nimmagadda et al. 2010).

The historical records of a given learner, including hardware capabilities, various learning preferences, and the current situation of wireless networks, can help solve the problems arising from variant mobile device capabilities, vacillating network conditions, and learners' diverse preferences. The proposed resolution is a concept that focuses on historical learners' requests to provide successful new requests, which share similar preference attributes. That is, if we provide a new request with the personalized learning content created from a previous similar request, not only will the performance of content delivery improve, but the learner's overall satisfaction will be greatly enhanced as well. This is complicated by the fact that there are vast amounts of educational data and information available for e-learners. In an e-learning system, teaching materials are usually stored in a Learning Object Repository (LOR), which can be formatted based on one of the most popular standards on the e-learning system, Sharable Content Object Reference Model (SCORM) (2010). In an LOR, a substantial amount of teaching materials, including related learning objects, will result in management issues over wired/wireless environments (Ko and Choy 2002; Wong et al. 2004; Su et al. 2005).

Therefore, in this paper, our main concern is how to efficiently manage a large number of historical learners' requests while delivering learning content that is precisely tailored to meet the individual needs of each user and to fall within the scope of various mobile device requirements. Successful management of similar content and delivery of that information to various mobile devices can be accomplished by providing learners with adapted content created from a similar, previously processed request. To achieve this end goal, the following must be realized:

(1) Model and define constructive and meaningful data in a format that is tailored to meet the individual needs and mobile requirements of each user.
(2) Group several historical learners' requests, which share similar preference attributes, into the same cluster to construct a useful relationship among them.
(3) Determine high-fidelity, personalized learning content quickly and efficiently for a new request based on the analysis of historical learners' requests.

While taking diverse learners' needs, wireless network conditions, as well as mobile device capabilities and constraints into consideration, a Personalized Learning Content Adaptation Mechanism (PLCAM) has been proposed. The proposed PLCAM can efficiently manage a large number of historical learners' requests, and intelligently deliver suitably personalized learning content, with higher fidelity, from a Learning Object Repository (LOR) directly to the learner. Subsequently, an adapted, or modified, version of the learning content would be prepared for the next similar request. Thus, the PLCAM is composed of two distinct phases: (1) the Adaptation Data Format Definition Phase and (2) the Personalized Learning Content Delivery Phase. The former phase defines an adaptation data format based on Composite Capabilities/Preference Profiles (CC/PP) (2010) and User Agent Profile (UAProf) (2010). These profiles identify constructive information about the requesting learner, applicable hardware, network capabilities, and media during the learning content adaptation process. The latter phase applies distance-based clustering and decision tree approaches to create a working relationship among a cache of historical learners' requests. At this stage, the proposed adaptation decision and content synthesis processes can resourcefully identify and organize a precise and suitable version of the learning content to meet the specific request.

Furthermore, the framework of the PLCAM can be extended to address many more diverse user needs and to enhance the effectiveness of the adapted content composition. In an effort to evaluate our proposed approach, a prototypical system of the PLCAM, based on a Sharable Content Object Reference Model (SCORM)-compliant LOR has been developed, and experiments have also been performed. The experimental results show that the PLCAM is efficient and can be expected to benefit learners.

2 Related work

Diverse content adaptation approaches have been proposed to render learning content on the mobile devices. Fudzee and Abawajy (2008) grouped content adaptation approaches into two basic types: static adaptation and dynamic adaptation. The former usually generates multiple variants for each content component, attaching a layout description for the presentation of component-based Web content (Mohan et al. 1999; Villard et al. 2000; Hinz et al. 2004). These static adaptation approaches can reduce download time, but they require preprocessing tasks and greater storage allocation. Another limitation is that they do not take into account the user's preference and the situation of the wireless network.

Many dynamic adaptation approaches, including content structure analysis and context-based adaptations, have been proposed to resolve these issues (Fudzee and Abawajy 2008). A Hierarchical Atomic Navigation Concept (HANd) was proposed by González-Castaño et al. (2002) to navigate on small-scale devices, using the content structure analysis approach. In the HANd approach, an automatically generated navigator page is used to indicate some or all elements embedded in a World Wide Web (WWW) page. To generate the navigator page, a Web page must be analyzed and fragmented into several separate "clipped" versions, which can be delivered to a small-scale device, according to an importance value levied for every page fragment. Based on a similar concept, many fragmentation and summarization processes have been proposed to organize a Web page into a thumbnail representation that indexes detailed information (Chen et al. 2003), breaks each Web page into several text units (Buyukkokten et al. 2001, 2002), and detects the important parts (Yin and Lee 2004) or the interesting fragments in dynamic Web pages (Ramaswamy et al. 2005), thus reducing delivery latency.

However, not all Web pages are suitable for text summarization because summarized statements, as lossy information, may mislead users. To help improve understanding, the semantically coherent perceivable units of the Web content can be extracted and presented together on a mobile device according to their semantic relationships (Laakko and Hiltunen 2005; Yang et al. 2007b; Kim et al. 2008). However, most of the aforementioned content adaptation approaches do not consider flexible and extensible content adaptation. Recognizing this, He et al. (2007) applied the rule-based approach to propose a flexible adaptation system, Xadaptor, for managing the new content type by adding its corresponding content adaptation rule. Similarly, a Versatile Transcoding Proxy (VTP) (Hsiao et al. 2008) executes the transcoding preference script provided by the client or server to transform the corresponding data or protocol according to the user's specification based on CC/PP (2010). Although most of the previously mentioned content structure analysis approaches can improve delivery latency, the quality of the content shown on mobile devices is limited to fixed page fragments, which are compatible with content adaptation processing approaches. These approaches, however, may experience loss of information, e.g., text and semantic relations, during the decomposition or summarization processes, and environmental context and user's preferences cannot yet be taken into consideration.

To consider the user's preference in the context-based adaptation approach, Lum and Lau (2003) proposed a decision engine, which can determine automatically an appropriate content adaptation version based on the Quality of Service-sensitive (QoS) approach. A typical score tree with several score nodes is created to evaluate the QoS of the content versions in various quality domains. Because they assume that user preferences and the adaptation process are independent of content, the score tree can be established during the preprocess phase. This predefined score tree with limited content versions may not meet the user's needs and may constrain the flexibility and extensibility of the system. Yang et al. (2008) took the learner's environmental context into account, proposing a context-driven content adaptation planner. This planner can determine a proper version of content created by static or dynamic adaptation, based on the predefined context profile to meet a learner's context status (e.g., brighten the background light in the outdoor environment). However, efficiently selecting and deciding upon an appropriate adapted version of content from previous requests is still an important and under-evaluated issue.

To take advantage of similar navigational behaviors, Mohomed et al. (2004) proposed an automatic content adaptation approach based on the community-driven concept, which artificially groups users into communities according to some similar characteristics and assumes that users in the same community have similar adaptation preferences. This system can learn the common adaptation preference of one community based on users' feedback. Later, a feedback-driven context selection approach (Mohomed et al. 2006b) was proposed to leverage user interaction to automatically split users into several groups according to feedback from their previous research system, URICA (Mohomed et al. 2006a). This system adapts content for mobile devices based on usage semantics. Using the same concept and detecting correlations in the adaptation requirements of past users, they initially applied the standard K-means clustering algorithm to partition users into multiple groups. They proposed the use of an online classification algorithm to make predictions that allowed rapid classification of each user into a single cluster, defined as suitable for offering a specific adapted content version to users (Mohomed et al. 2007). Their experimental results showed that bandwidth consumption and browsing time can be significantly reduced. However, this approach clusters users according to the fidelity of accessed image objects and must refine the prediction of adapted content by means of the user's interaction and feedback. According to existing research (Muntean 2008; Ding et al. 2010), the long waiting and response time can explicitly affect a user's satisfaction, often leading to negative feedback. Therefore, if the desired adapted content is received by several interactions, a poor navigation experience may result due to the long waiting time. Lee et al. (2006) also proposed an intelligent adaptation system to reduce response time by classifying users into basic categories. This process can be adjusted according to the users' feedback; content generated by the same group category can be reused.

However, in learning environments, the learners' preferences vary greatly because each learner is an individual student, and thus, has his or her individualized learning behaviors and preferences. The variety of behaviors and preferences can include content type preferences such as text, picture, audio, video, or a hybrid of those types (Gilbert and Han 1999; Stern and Woolf 2000). Learners also differ in their inclinations for how learning material is presented and how they intellectually receive and digest that material. That is, learners vary in their preferences of presentation styles, cognitive styles (field dependence/independence) (Riding and Cheema 1991), and learning styles (such as whether they are "doers," "watchers," "thinkers," or "feelers") (Kolb 1976, 2004). Students will achieve higher learning performance if the learning content can be customized and offered according to their diverse learning needs (Beekhoven et al. 2003; Chen and Mizoguchi 1999; Dewhurst et al. 2000; Chen et al. 2000; Smith 2001; McIlroy et al. 2001; Riding and Cheema 1991; Wilson 2000; Su et al. 2005). Recognizing these realities in mobile learning environments, Tong et al. (2006) used predefined adaptation rules to adapt the learning content to meet learning preference and environmental conditions. Franzoni et al. (2008) proposed learning styles integration taxonomy to facilitate and personalize the learning process. Nevertheless, the predefined approach is time consuming and to the efficient use of the adaptation rule is still problematic. In addition, response time of the content adaptation system is an important factor that affects the user's satisfaction because it measures waiting time (Ding et al. 2010). Muntean (2008) proposed a learner Quality of Experience (QoE) model in an educational area derived from delivery performance-based content personalization to investigate the influence of delivery latency on user experience and learning performance. The experimental results demonstrate the significant benefits in learning achievement, performance, and satisfaction when learners are offered superior content quality and delivery performance. In other words, learners' satisfaction and performance are affected not only by their perceptions and preferences, but also by network conditions and mobile device factors in mobile learning environments. Nevertheless, most existing research and systems either separately or partially considered the features of mobile devices, network conditions, and learner preferences (Zhao et al. 2008). Consequently, they may not be suitable for efficient management of diverse user needs in mobile learning environments.

3 Personalized learning content adaptation mechanism

3.1 Framework of the personalized learning content adaptation mechanism

This paper proposes a **PLCAM**, to address the increasing number and diversity of user requests. The architecture of the PLCAM is shown in Fig. 1. The PLCAM can manage efficiently a large number of historical learners' requests and intelligently deliver proper personalized learning content with higher fidelity from a Learning Object Repository (LOR) directly to the learner. An adapted content version can then be prepared for the next similar request.

The PLCAM includes two phases, as described below:

1. *Adaptation Data Format Definition Phase*: First, we define the adaptation data format including Learner Preference (**LP**), Hardware Profile (**HP**), and Media

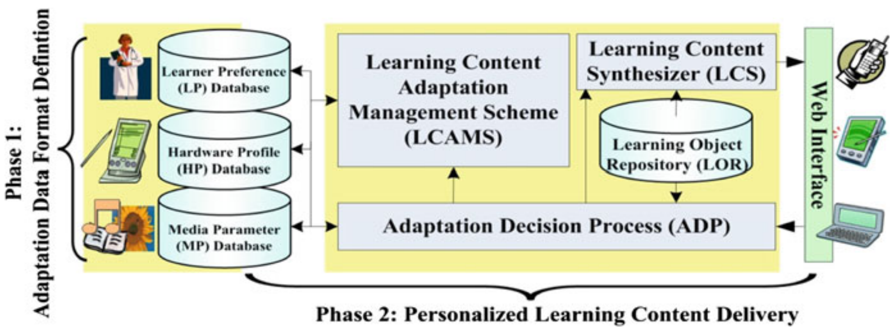


Fig. 1 Architecture of PLCAM

Parameter (**MP**) based upon CC/PP (2010) and UAProf (2010). Here, the **LP** describes the diverse needs of a learner, e.g., the desired maximum delivery time, image format, presentation style, the ratio of audio to picture, etc. The **HP** describes the hardware capabilities of a learner's mobile device, e.g., the device type, screen size, etc. The **MP** describes how to adapt and transcode the multimedia, e.g., image, audio, and video, within requested learning objects.

2. *Personalized Learning Content Delivery Phase*: To deliver the suitable learning content with associated learning resources efficiently to learners in accordance with their preferences, their device's hardware capabilities, and variable wireless bandwidth, we propose the following three modules.
   - *Learning Content Adaptation Management Scheme (LCAMS)*: To manage the learners' historical request data efficiently, we first apply the distance-based clustering approach to group the historical learners' requests into several sets, according to Learner Preference (LP). After the initial clustering phase, every cluster created with a similar LP will be tagged with a cluster label. The Hardware Profiles (HP) within the historical requests with corresponding cluster labels are used as training data to create a decision tree, called Content Adaptation Decision Tree (CADT). The CADT can be used to efficiently determine the appropriate adapted content.
   - *Adaptation Decision Process (ADP)*: To determine a precise and proper adapted content version based on CADT, we propose an ADP Algorithm, called ADPAlgo, which can determine a suitable version of the existing adapted content.
   - *Learning Content Synthesizer (LCS)*: According to the results of ADPAlgo, the LCS will use the identified adaptation parameters to transcode the content if necessary.

The details of each phase are described in the following subsections.

3.2 Adaptation data format definition

To manage existing user requests efficiently, we must model and define a data format including the LP, HP, network condition, and MP, which will be recorded in a database to represent every learner's request based upon CC/PP (2010) and UAProf (2010) (Kobsa 2007). Thus, a Content Adaptation Rule (CAR) is defined to represent a processed learner request transaction in the PLCAM.

Definition 1 Content Adaptation Rule (CAR)
$\mathbf{CAR} = (\mathbf{LO}_i, \mathbf{p}_j, (\mathbf{B}, \mathbf{HP}, \mathbf{LP}), \mathbf{MP}_{set})$, where:

- $\mathbf{LO}_i$: the ith learning object in the SCORM-compliant LOR.
- $\mathbf{p}_j = \{r_1, r_2, \ldots, r_m\}$: the j_{th} page of $\mathbf{LO}_i$, i.e., $(\mathbf{LO}_{ij})$ consists of several associated learning resources (r).
- **B**: the bandwidth of the network condition in a mobile learning environment during the learner request.
- $\mathbf{HP} = \langle a_1, a_2, \ldots, a_n \rangle$: every attribute (a) denotes a specific capability of a mobile device, e.g., the machine type (PDA or smartphone), Central Processing Unit (CPU) speed, memory capacity, screen size, sound rate, etc.

- $\mathbf{LP} = \langle b_1, b_2, \ldots, b_k \rangle$: every attribute (b) denotes a learner's specific requirement, e.g., maximum delivery time, preferred picture format ordering, preferred audio property, media switch, preferred content type, cognitive style, learning style, etc. Thus, in this paper, we can initially define the LP = ⟨*Delivery Time* (DT), *Preferred Picture Format Ordering* (PPFO), *Picture Switch* (PS), *Audio Switch* (AS)⟩ described in Example 1. The preferred content type, cognitive style, and learning style can also be extended into LP definition if necessary, and the details are described in Sect. 6.3, The Extensibility of PLCAM.
- $\mathbf{MP}_{set} = \{\mathbf{MP}_1, \mathbf{MP}_2, \ldots, \mathbf{MP}_k\}$: denotes all associated Media Parameter (MP) used to adapt and transcode all physical media resources belonging to a page. Here, MP = ⟨*Version* (**V**), *Type* (**T**), *Attribute* (**A**), *Size* (**S**), TP⟩, where TP is the *Transcoding Parameters* used by the existing transcoding tools or approaches, e.g., ImageMagicK (2010) and SoX (2010), to transform the original media object **($\mathbf{V}_0$)** into adapted media version ($\mathbf{V}_i$) with type **(T)**, e.g., image, audio, video, associated with media attributes **(A)**, e.g., image = ⟨width, height, color depth, type⟩, audio = ⟨precision, rate⟩, video = ⟨width, height, sound precision, sound rate⟩, and file size (**S**). In the PLCAM, the media transcoding process will be triggered if there is no suitable adapted content version for a new learner's request.

Example 1 A CAR of a given learner's request transaction can be recorded as $((\mathrm{LO}_i, \mathrm{p}_j)$, (80, HP, LP), $\mathrm{MP}_{set})$, where **HP** = ⟨1,400, 128, 480, 640, 16, 16, 44⟩ denotes that a learner uses a PDA (**1**) with **400** Mhz, **128** MB, **480** × **640** resolution, **16** bits color depth, **16** bits sound precision and **44** KHz sound rate (**U** denotes Unsupported) under **80** kbps bandwidth (**B**) to retrieve the page ($\mathbf{p}_j$) of $\mathbf{LO}_i$, and **LP** = ⟨5, JPGB, 1, 0⟩ denotes that the maximum delivery time (**DT**) is less equal than **5** seconds (s), the order of preferred picture format (**PPFO**) is JPG (**J**) > PNG (**P**) > GIF (**G**) > BMP (**B**), the switch attribute of media, **PS** = 1, enables to show the picture, and the **AS** = 0 disables the audio play, respectively. Then, the PLCAM is able to use the MPs in MP_{set}, which are selected according to the B, HP, and LP, to transcode the physical resources in the p_j. Table 1 shows the example with 15 CARs for the p_j in LO_i, i.e., LO_{ij}. The attribute definitions of LP and HP can be extended to meet the various requirements.

4 Learning content adaptation management scheme

In this section, we will describe how to use existing CARs to construct a Content Adaptation Decision Tree (CADT) in the Learning Content Adaptation Management Scheme (LCAMS). The CADT can be used to efficiently and quickly determine the suitable adapted content in an LOR for learners according to the mobile device features, the preferences of learners, and network bandwidth. As shown in Fig. 2, the LCAMS includes three processes to construct the CADT: (1) clustering process; (2) decision tree construction; and (3) CADT maintenance process.

4.1 Clustering process of the learning content adaptation management scheme

As mentioned in Sect. 3, each learner's request, with his/her preference(s) logged as a transaction, can be represented by a Content Adaptation Rule (CAR). As shown in

Table 1 Example of CAR for the same page (p_j) in an LO (LO_i), (LO_{ij})

ID	Bandwidth (B)	Hardware profile (HP)	Learner preference (LP)
1	213	⟨2, 528, 384, 320, 480, 16, 32, 120⟩	⟨7, JBGP, 1, 1⟩
2	175	⟨2, 600, 384, 320, 480, 24, 16, 40⟩	⟨2, GPBJ, 0, 1⟩
3	487	⟨0, 1200, 4000, 1366, 768, 32, 16, 300⟩	⟨2, JGPB, 1, 0⟩
4	223	⟨2, 528, 288, 320, 480, 16, 8, 30⟩	⟨3, JBPG, 1, 0⟩
5	281	⟨2, 528, 384, 320, 480, 16, 32, 120⟩	⟨7, PJGB, 0, 1⟩
6	69	⟨0, 2000, 8000, 1366, 768, 32, 32, 500⟩	⟨1, GPBJ, 1, 0⟩
7	232	⟨2, 528, 288, 320, 480, 16, 8, 30⟩	⟨5, JPGB, 0, 1⟩
8	290	⟨1, 1000, 448, 480, 800, 24, 16, 140⟩	⟨1, GPJB, 1, 0⟩
9	95	⟨0, 1200, 4000, 1366, 768, 32, 16, 300⟩	⟨1, BJGP, 0, 0⟩
10	167	⟨0, 1200, 4000, 1366, 768, 32, 16, 300⟩	⟨5, GPJB, 1, 0⟩
11	220	⟨0, 1200, 4000, 1366, 768, 32, 16, 300⟩	⟨5, JPGB, 0, 1⟩
12	326	⟨2, 528, 288, 320, 480, 16, 8, 30⟩	⟨4, JPGB, 0, 0⟩
13	339	⟨2, 528, 288, 320, 480, 32, 8, 20⟩	⟨7, PBGJ, 0, 0⟩
14	313	⟨2, 528, 288, 320, 480, 32, 8, 20⟩	⟨4, GJPB, 1, 1⟩
15	95	⟨2, 528, 384, 320, 480, 16, 32, 120⟩	⟨4, PBJG, 0, 0⟩

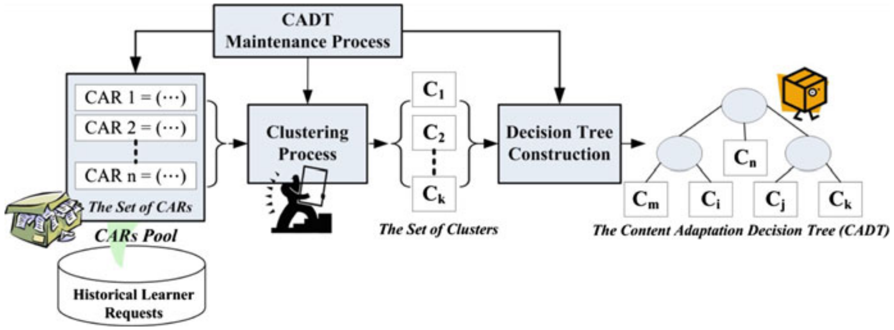


Fig. 2 Process of LCAMS

Fig. 2, in the LCAMS, all new CARs requested by learners will be stored temporarily in a CAR Pool. We can apply the distance-based clustering algorithm to group these historical CARs into several clusters according to learners' preferences, where every learner in the same cluster shares similar LPs (Tan et al. 2005; Romero and Ventura 2006, 2010). However, it is difficult to determine the number of clusters while applying the clustering approach. To resolve this problem and equip the PLCAM with the automatic maintenance process, the renowned clustering algorithm, ISODATA (Hall and Ball 1965), can be employed. This process can dynamically change the number of clusters by lumping and splitting procedures and by iteratively changing the number of clusters to produce better results. The ISODATA clustering approach has been used successfully in many applications, such as image processing (Rahimi et al. 2007; Chen et al. 2009), data and document classification (Zhu et al. 2007), and so on. In this study, we apply the ISODATA clustering approach to group CARs into different clusters automatically.

4.1.1 Similarity measure of the clustering process

To apply the ISODATA clustering approach, a similarity measure estimating the similarity value between two CARs based on the LP must be determined. Because the attribute of an LP might consist of a *numerical attribute*, e.g., *maximum delivery time*, and a *symbolic attribute*, e.g., *preferred picture format ordering*, as described in Example 1, the similarity measure of an LP can be formulized by means of the distance measure approach as follows:

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

$$SimofNum_k = 1 - \frac{|a_k - b_k|}{Max_k - Min_k},$$

where $1 \leqq k \leqq n$, the Max_k and the Min_k are the predefined maximum and minimum values of k_{th} attribute in an LP, respectively.

Regarding the symbolic attribute in an LP, the value, *JPGB*, is like a *string*. To calculate the similarity between two symbolic attributes, their string-based values can be encoded into a numerical value by the numerical order of predefined symbol priority. For instance, the numerical value of string *JPGB* can be encoded as ‘1234’ based on the priority order definition {“J” = 1, “P” = 2, “G” = 3, “B” = 4}. Also, the maximum value will be the ‘4321’ of the string *BGPJ*. The symbolic attribute can thus be transformed into a numerical value and its similarity can be measured by the $SimofNum_k$ formula. Because attributes in an LP may have different degrees of importance, we define a *Weight Vector* (*WV*), which can be manually defined by a learner, to adjust for the degree importance of each attribute. Therefore, the similarity measure between two LPs can be formulated as:

$$Similarity_{LP}\left(\mathrm{LP}_i,\ \mathrm{LP}_j\right) = \Sigma\left(SimofNum_k\left(\mathrm{a}_k,\ \mathrm{b}_k\right) \times w_k\right),$$
$$\text{where } w_k \in WV,\ \Sigma w_k = 1, \quad \text{and} \quad 1 \leqq k \leqq n.$$

To evaluate when to split and merge the cluster, the $Deviation_{LP}$, which is used to calculate the standard deviation of the samples, must be defined as:

$$Deviation_k^{LP} = |\frac{a_k - b_k}{Max_k - Min_k}|,,$$

where$1 \leqq k \leqq n$, the $Max_k$and the Min_k are the predefined maximum and minimum values of k_{th} attribute in an LP, respectively.

Example 2 Given two LPs, $\mathrm{LP}_1 = \langle 3,\ \mathrm{JPGB},\ 1,\ 0\rangle$ and $\mathrm{LP}_2 = \langle 2,\ \mathrm{JGBP},\ 1,\ 1\rangle$, and a learner predefined related attribute $\mathrm{WV} = \langle 0.5, 0.3, 0.1, 0.1\rangle$. We can apply the above similarity measure to calculate the similarity between LP_1 and LP_2. For example, the similarity of the numerical attribute, *Delivery Time* (DT), between LP_1 and LP_2 is:

$$SimofNum_1 = 1 - \frac{|a_1 - b_1|}{Max - Min} = 1 - \left(\frac{|3-2|}{5-1}\right) = 1 - \frac{1}{4} = 0.75.$$

The similarity of the symbolic attribute, Preferred Picture Format Ordering (PPFO), is:

$$SimofNum_2 = 1 - \left(\frac{|1234 - 1423|}{4321 - 1234}\right) = 1 - \frac{189}{3087} = 0.938.$$

Hence, by the same way, the similarity between LP_1 and LP_2 is the $Similarity_{LP}(LP_1, LP_2) = 0.75 \times 0.5 + 0.938 \times 0.3 + (1 - (|1 - 1|/(1 - 0))) \times 0.1 + (1 - (|0 - 1|/(1 - 0))) \times 0.1 = 0.7564$. Besides, the $Deviation_1^{LP} = \left|\frac{3-2}{5-1}\right| = 0.25$.

Algorithm 1

Algorithm 1: LP Clustering Algorithm (LPCALgo)

Symbols Definition:

DT: the Delivery Time (DT) in a learner preference vectors (LP).

$\mathbf{LP}_{set}$: the set of LP.

K: the initial number of clusters.

C: a cluster with several learner preference vectors (LP).

CC: the Center of Cluster.

$\mathbf{C}_{set}$:the set of clusters with the Center of Cluster (CC)

$\mathbf{T}_s$:the split threshold (Standard Deviation) for splitting a cluster into two ones.

$\mathbf{T}_m$:the merge threshold (Mean Distance) for merging two clusters into one.

$\mathbf{T}_n$:the minimum number of the members in a Cluster for deleting a cluster.

$\mathbf{T}_i$: the maximum iteration number for executing the clustering process

$\mathbf{T}_p$:the minimum number of Cluster pair for merging clusters process.

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

Output: The set of Clusters, $\mathbf{C}_{set}$.

Step 1: Initial Clusters Selection:

Step 1.1: *For i* = 1 to **K**.

(**1**) random select $\mathrm{LP}_i \in \mathrm{LP}_{set}$ to insert LP_i into C_i with $\mathrm{CC}_i = \mathrm{LP}_i$ and then insert C_i into $\mathbf{C}_{set.}$

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 or exceed the $\mathbf{T}_i$.

Step 2.2: Insert each $\mathrm{LP}_j \in \mathbf{LP}_{set}$ into appropriate cluster $C_i \in \mathbf{C}_{set}$ according to the $Similarity_{LP}(\mathrm{CC}_i, \mathrm{LP}_j)$.

Step 2.3: Delete the C_i if the number of LP is less than $\mathbf{T}_n$.

Step 2.4: Split a C_i into two clusters according to the $\mathbf{T}_s$ and $\mathbf{T}_n$.

Step 2.5: Merge two clusters into one according to the $\mathbf{T}_m$ and $\mathbf{T}_p$.

Step 2.6: Re-compute the Cluster Center (CC_i) for each $C_i \in \mathbf{C}_{set}$.

Step 3: Output the $\mathbf{C}_{set}$.

4.1.2 Clustering algorithm based on ISODAT

An *LP Clustering Algorithm* (LPCALgo) based on ISODATA is proposed to group these LPs into several clusters according to the aforementioned similarity and deviation measure, shown in Algorithm 1. After applying the LPCALgo, the CARs in Table 1 can be grouped into three clusters, as depicted in Table 2.

Table 2 Result of applying LP clustering algorithm (LPCALgo) with the cluster parameters (K = 5, $T_s = 0.01$, $T_m = 1.0$, $T_n = 1$, $T_i = 50$, $T_p = 1$) based on data in Table 1

Cluster label	ID of CAR
1	{3, 4, 6, 8, 10}
2	{2, 5, 7, 9, 11, 12, 13, 15}
3	{1, 14}

Table 3 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

4.2 Decision tree construction

After the clustering process, each cluster will be tagged with a label, as shown in Table 2. Determining a suitable cluster for a new request is an issue which can be resolved by using the decision tree approach. Based on the Hardware Profiles (HPs) in these CARs, with cluster labels defined in Table 2, we can apply a decision tree induction algorithm, ID3 (Quinlan 1986), to create a decision tree, called CADT. ID3 can process only the symbolic value of an attribute, so the numerical attribute values of the HP in Table 1, e.g., CPU speed, system memory, etc., have to be discretized by the following approach.

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

Therefore, the numerical attribute of HP in Table 1 can be mapped into several discrete ranges, as shown in Table 3.

In ID3, the information gain measure is used to select the test attribute at each node in the decision tree. The attribute with the highest information gain is chosen as the test attribute for the given set (Quinlan 1986). The following example describes how to apply ID3 to create the CADT of the PLCAM based on the HPs in the CARs with cluster labels.

Example 3 Table 4 shows six HP data with an ID and cluster label, which have been classified into two subsets: {4, 7, 12} and {5, 15, 1} according to the attribute, "*Sound Precision.*" The expected information needed to classify six samples is given by I (the number of CARs in C_1, the number of CARs in C_2, the number of CARs in C_3) $= I(1,\ 4,\ 1) = -\left(\frac{1}{1+4+1}\right)\log_2\left(\frac{1}{1+4+1}\right) - \left(\frac{4}{1+4+1}\right)\log_2\left(\frac{4}{1+4+1}\right) - \left(\frac{1}{1+4+1}\right)\log_2\left(\frac{1}{1+4+1}\right) = 1.252.$

Table 4 HP in CAR data with ID and cluster label classified by attribute, "*Sound Precision*"

ID	HP in CAR	Cluster label
4	⟨2, 528, 288, 320, 480, 16, **8**, 30⟩	1
7	⟨2, 528, 288, 320, 480, 16, **8**, 30⟩	2
12	⟨2, 528, 288, 320, 480, 16, **8**, 30⟩	2
5	⟨2, 528, 384, 320, 480, 16, **32**, 120⟩	2
15	⟨2, 528, 384, 320, 480, 16, **32**, 120⟩	2
1	⟨2, 528, 384, 320, 480, 16, **32**, 120⟩	3

The Entropy (E), or expected information based on the partitioning into two subsets by the attribute, "Sound Precision," is given by:

$$\begin{aligned} &E(Sound Precision)\\ &= \frac{(1+2)}{(1+4+1)} I(1,2) + \frac{(2+1)}{(1+4+1)} I(2,1)\\ &= \frac{3}{6}\left(-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}\right) + \frac{3}{6}\left(-\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3}\right)\\ &= 0.918. \end{aligned}$$

Finally, the encoding information that would be gained by branching on attribute "*Sound Precision*" is Gain(*Sound Precision*) $= I(1,4,1) - E(Sound Precision) = 1.252 - 0.918 = 0.334$.

Consequently, by means of the above ID3 approach, the information gain of each attribute (of each HP) in Table 4 will be computed. The attribute with the highest information gain will be chosen as the test attribute. A node is created and labeled with the attribute, branches are created for each value of the attribute, and the samples are partitioned accordingly. Figure 3 depicts the result of applying the ID3 algorithm data given in Tables 1, 2, and 3.

4.3 Content adaptation decision tree maintenance process

As stated previously, after the clustering and decision tree construction processes are complete in the LCAMS, all CARs in the CAR Pool, a temporary buffer, can be grouped into several clusters and retrieved by the CADT structure. In the CADT maintenance process (see Fig. 4), all new CARs are first temporarily stored in a CAR Pool. While the amount of CARs (N) in a CAR Pool is more than a threshold, which is estimated automatically by the *CADT Rebuilding Equation* ($Y = \alpha + \beta X$) generated by the ordinary least squares approach and described in Sect. 6, the LCAMS will rebuild the CADT automatically offline by the clustering and decision tree processes. Then, these processed CARs in the CAR Pool will be shifted to the final storage and become the historical CARs indicated by the newly rebuilt CADT structure. Each CAR indicates the associated media objects consisting of original (V_0) or adapted versions

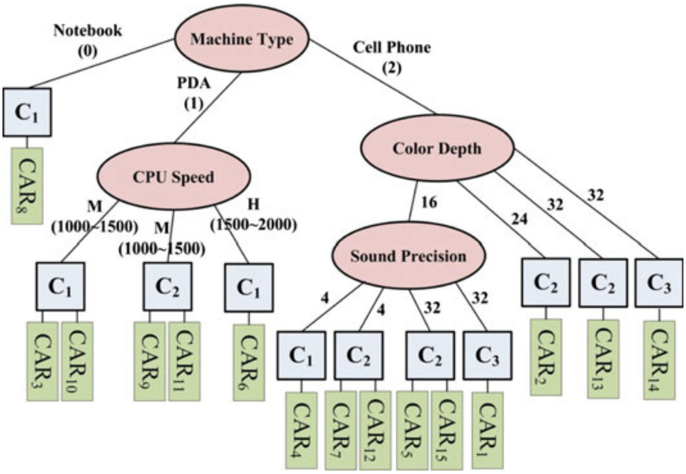


Fig. 3 CADT based on the HP in Table 1

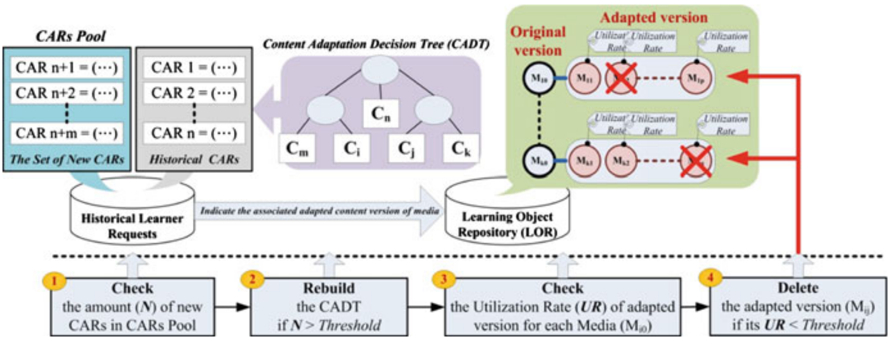


Fig. 4 Flowchart of the CADT maintenance process

$(V_i|i > 0)$, all of which are stored in the LOR. Moreover, in order to efficiently manage the storage space of the LOR, the LCAMS will check the Utilization Rate (*UR*) of every adapted media object version, except for its original version. If the *UR* of any adapted version $(M_{ij}) <$ Threshold, it will be deleted from the LOR.

5 Personalized learning content delivering process

5.1 Content adaptation process

To meet diverse learner needs, including varied mobile device capabilities, network conditions, and individual learner preferences, the Content Adaptation Process (CAP) has been proposed to automatically determine an appropriate MP_{set} from the Media Parameter (MP) database to adapt and transcode all media resources in a desired page according to the requirement of the new Learner Request (LR). The process is described below.

5.1.1 Satisfaction measure on the quality of media

In the CAP, we would like to determine an adapted media which can meet the requirement of an LR very well. Therefore, the satisfaction measure on the quality of media has been defined to estimate the satisfaction degree between the adapted media selected by CAP and the media requested by the learner.

Given HP = $\langle a_1, a_2, \ldots, a_m \rangle$ and $MP_i = \langle b_1, b_2, \ldots, b_n \rangle$, the similarity measure of each numerical attribute between HP and MP_i can be formulized as:

$$\mathit{SimofNum}\,(\mathrm{HP}, \mathrm{MP}_i) = Max\left(1 - \frac{|a_j - b_k|}{a_j}, 0\right), \text{ where} 1 \leqq j \leqq m, 1 \leqq k \leqq n.$$

Regarding the symbolic attribute of the image type between the MP and the requested LP, e.g., *Preferred Picture Format Ordering (PPFO)*, the particular similarity measure of image type is formulized as:

$$\mathit{SimofImage}_{Type}(\mathrm{PPFO} \in \mathrm{LP},\ \mathrm{Type} \in \mathrm{MP}_i) = 1 - (\mathrm{k} - 1) \times 0.25,$$
$$\text{where k} = \text{the order of type in the string of PPFO.}$$

We also define a *Satisfaction Weight Vector* (*SWV*) to adjust the degree of importance. The satisfaction measure on the quality of media between an LR and an MP in the media database can thus be formulated as:

$$\begin{aligned}&Satisfaction_{QualityOfMedia}\,(\mathrm{LR},\ \mathrm{MP}_i)\\&= \Sigma((\mathit{SimofNum}(a_j \in \mathrm{HP}, b_k \in \mathrm{MP})|\mathit{SimofImage}_{Type}\\&\quad(\mathrm{PPFO} \in \mathrm{LP},\ \mathrm{Type} \in \mathrm{MP})) \times w_j),\\&\quad\text{where} w_j \in SWV, \Sigma w_j = 1, \quad\text{and}\quad 1 \leqq j \leqq m, 1 \leqq k \leqq n.\end{aligned}$$

Example 4 Given $\mathbf{MP}_a = \langle$*Version* **(V),** *Type*(**T**)**,** *Attribute*(**A**)**,** *Size*(**S**), TP$\rangle = \langle \mathbf{v}_1$, image, $\langle$800, 400, 16, J$\rangle$, 160, TP$\rangle$ with *SMV* = $\langle$0.5, 0.1, 0.3, 0.1$\rangle$, and a new **LR** = $((\mathrm{LO}_i, \mathrm{p}_j)$,(B, HP, LP)) = $((\mathrm{LO}_1, \mathrm{p}_1)$, (500, $\langle$1, 1000, 448, **480**, **800**, **24**, 16, 140$\rangle$,$\langle$1, **GPJB**, 1, 0$\rangle$))). Then, the satisfaction between LR and MP_a can be estimated as follows:

$$\begin{aligned}&Satisfaction_{QualityOfMedia}\,(\mathrm{LR},\ \mathrm{MP}_a) = 0.5 \times \mathit{SimofNum}(480 \in \mathrm{HP}, 800)\\&+0.1 \times \mathit{SimofNum}(800 \in \mathrm{HP}, 400) + 0.3 \times \mathit{SimofNum}(24 \in \mathrm{HP}, 16) + 0.1\\&\times \mathit{SimofImage}_{Type}(\text{“GPJB”} \in \mathrm{LP}, \text{‘J’}) = 0.5 \times \mathrm{Max}\,(0.33, 0) + 0.1 \times \mathrm{Max}\,(0.5, 0)\\&+0.3 \times \mathrm{Max}\,(0.66, 0) + 0.1 \times (1 - (3-1) \times 0.25) = 0.165 + 0.05 + 0.198\\&+ 0.05 = 0.463.\end{aligned}$$

5.1.2 Satisfaction score of the media parameter

By means of the $Satisfaction_{QualityOfMedia}(\mathrm{LR}, \mathrm{MP}_i)$, we can understand which adapted media is more suitable to meet the requirements of a given LR. However, as mentioned in Sect. 1, the response time to LRs will explicitly affect learner satisfaction (Muntean 2008; Ding et al. 2010). Accordingly, we take the response time of the PLCAM into account and define the satisfaction score of the MP to estimate the satisfaction degree of applying the MP to adapt the Media Object (MO), whereby the most appropriate $\text{MP}_{set} = \{\text{MP}_1, \text{MP}_2, \ldots, \text{MP}_k\}$ can be determined by the CAP. The definition of the satisfaction score is as follows and the CAP algorithm is described in Algorithm 2:

$$SatisfactionScore_{MP}\left(Satisfaction_{\text{QualityOfMedia}}\left(\text{LR}, \text{MP}_i\right), \text{T}_{expected}, \text{T}_{used}\right) = \begin{cases} \frac{Satisfaction_{\text{QualityOfMedia}}}{1+(\text{T}_{used}-\text{T}_{expected})/\text{T}_{expected}}, & \text{if } \text{T}_{used} > \text{T}_{expected}, \\ Satisfaction_{\text{QualityOfMedia}}, & \text{Otherwise} \end{cases}$$

where $\text{T}_{expected}$ = the maximum available DT and T_{used} = the actual time spent delivering this adapted media Version (**V**) transcoded by MP_i.

Example 5 Given an $\mathbf{LR} = ((\text{LO}_1, \text{p}_1), (500, \langle 1, 1000, 448, 480, 800, 24, 16, 140\rangle, \langle 1, \text{GPJB}, 1, 0\rangle))$. Assume there are two MOs, $\text{MO}_1 = \text{Image}$, $\text{MO}_2 = \text{audio}$, in requested page p_1 of LO_1, where MO_1 has an $\text{MP}_1 = (\text{“MO}_{1v1}\text{”}, \text{image}, \langle 800, 400, 16, \text{J}\rangle, 160)$ without a corresponding physical adapted media file in the LOR.

Therefore, the MO_1 will be added into $\mathbf{Media}_{req}$ only due to the Audio Switch (AS) is 0, i.e., $\mathbf{Media}_{req} = \{\text{MO}_1\}$ **(Step 1)**. Then, all MPs of MOs in $\mathbf{Media}_{req}$ will be inserted into $\mathbf{MP}_{candi}$ for calculating the *satisfaction score*. Thus, the $\mathbf{MP}_{candi} = \{\text{MP}_1\}$ **(Step 2)**. Afterwards, we can estimate the $\mathbf{T}_{expected} = 1/1 = 1$ to understand how much time we can use to do the CAP for each requested MO **(Step 3)**.

For each $\text{MP}_i \in \text{MP}_{candi}$, we estimate how much time we need to spend delivering the media size over the Bandwidth (B) of the wireless network, i.e., $\mathbf{T}_{deliver} = 160/500 = 0.32$ s; and how high the $Satisfaction_{\text{QualityOfMedia}}$ is, i.e., $\mathbf{Sat}_1 = 0.463$, as described in Example 4. Then, because MP_1 has no corresponding physical media file in the LOR, the nearest physical media file, $(\text{“MO}_{1v2}\text{”}, \text{image}, \langle 1000, 500, 16, \text{J}\rangle, 160)$, in the LOR will be selected to estimate its transcoding time in advance if we deliver it to the user. Here, we can assume $\mathbf{T}_{transcoding} = 1$ s. Therefore, the satisfaction score of MP_1 can thus be calculated by:

$$SatisfactionScore_{MP}(0.463, \mathbf{1}, 0.32 + \mathbf{1}) = \frac{0.463}{1 + (1.32 - 1)/1} = 0.35. \quad \textbf{(Step4)}$$

Finally, if the MP_1 has the maximum satisfaction score in terms of MO_1, it will be selected to insert into MP_{set}, which will be used to perform the learning content synthesis, as described in Sect. 5.3 **(Steps 5 and 6)**.

Algorithm 2

Algorithm 2: Content Adaptation Process Algorithm (CAPAlgo)
Symbol Definition:
LR: denotes a learner request, i.e., LR = (LO, (B, HP, LP)).
MO: denotes a Media Object $\in$ LO
$\mathbf{MP}_{candi}$:the candidate MP list
$\mathbf{Media}_{req}$: the set of requested media object
$\mathbf{MP}_{set}$={MP_1, MP_2,..., MP_k}: stores all appropriated MPs selected by CAPAlgo.
$\mathbf{Sat}_i$: the $Satisfaction_{QualityOfMedia}$ of $MP_i = \langle V_i, T, A, S, TP\rangle$
$\mathbf{T}_{MDT}$: maximum available delivery time, default is DT $\in$ LP in LR
$\mathbf{T}_{expected}$:the average expected time of deivering each requested media object
$\mathbf{T}_{deliver}$: the estimated deliver time of the media version (V) in MP_i
$\mathbf{T}_{transcoding}$: the estimated transcoding time of the media version (V) in MP_i
Input: a **LR,** $\mathbf{T}_{MDT}$
Output: $\mathbf{MP}_{set}$
Step 1: add all requested media objects (MOs) $\in$LO in LR into $\mathbf{Media}_{req}$
Step 2: for each MO $\in$ $\mathbf{Media}_{req}$, add all MP_i $\in$MO into MP_{candi}
Step 3: calculate$\mathbf{T}_{expected} = \mathbf{T}_{MDT}$/(the number of MO in $\mathbf{Media}_{req}$)
Step 4: for each $MP_i \in MP_{candi}$
(1) Calculate $\mathbf{Sat}_i = Satisfaction_{QualityOfMedia}(\text{LR}, MP_i)$
(2) Calculate $\mathbf{T}_{deliver} = size(S) \in MP_i / Bandwidth(B) \in \text{LR}$
(3) Calculate $\mathbf{T}_{transcoding} =$
$$\begin{cases} 0, if\ physical\ file\ of\ MP_i\ is\ in\ Learning\ Object\ Repository\ (LOR) \\ Estimate\ transcoding\ time\ from\ nearest\ physical\ file\ in\ LOR \end{cases}$$
(4) Calculate Satisfaction Score of $MP_i = SatisfactionScore_{MP}(\mathbf{Sat}_i, \mathbf{T}_{expected}, \mathbf{T}_{deliver} + \mathbf{T}_{transcoding})$
Step 5: for each MO $\in$ $Media_{req}$,
(1) Select MP_i with maximum Satisfaction Score and store it into MP_{set}
Step 6: Return the MP_{set}.

5.2 Adaptation decision process

In the Learning Content Adaptation Management Scheme (LCAMS), the Content Adaptation Decision Tree (CADT) can be used to search, retrieve, and maintain historical CARs. The desired adapted contents can be delivered quickly to learners if there is a similar existing learner request held by CADT. Determining how to efficiently deliver an appropriate adapted content from the existing CARs or how to redo the aforementioned Content Adaptation Process (CAP) is a concern. We propose an *Adaptation Decision Process Algorithm* (ADPAlgo) to process the adapted content decision quickly. The ADPAlgo is shown in Algorithm 3 and illustrated in Fig. 5.

In the Adaptation Decision Process (ADP), as shown in Fig. 5, we are given a new LR = ((LO_i,p_j), (B_{LR}, HP_{LR}, LP_{LR})). First, a suitable cluster will be selected by traversing the CADT based on HP_{LR}. Second, these CAR = ((LO_k,p_m), (B_{CAR}, HP_{CAR}, LP_{CAR})) in the selected cluster will be merged with those in the CAR Pool, shown in Fig. 2. Third, a CAR will be deleted if it satisfies one of four selection rules, e.g., $(\mathrm{LO}_i,\mathrm{p}_j) \neq (\mathrm{LO}_k,\mathrm{p}_m)$. Fourth, if there is a remaining CAR with higher similarity compared to the LR, the Learning Content Synthesizer (LCS) in Sect. 5.3 will compose the personalized learning content and transcode the associated MOs based on necessity. Then, the adapted learning content will be delivered to a learner directly without or with low transcoding latency. Otherwise, the CAP will be triggered to create a new CAR based on the LR.

Algorithm 3

Algorithm 3: Adaptation Decision Process Algorithm (ADPAlgo)
Symbol Definition:
CAR$_{set}$: stores several historical CARs
LR: denotes a learner request, i.e., LR = (LO, (B, HP, LP)).
CAR$_{new}$: stores the new CAR created according to the LR.
$\boldsymbol{\alpha}$: denotes the acceptable percent threshold of bandwidth deviation.
$\boldsymbol{\beta}$: denotes the acceptable weight Threshold of the amount (N_{HP}) of attributes in HP.
$\boldsymbol{\gamma}$: denotes the acceptable threshold of Similarity value.
$\mathrm{S}_{Min} = \beta \times N_{HP}$: denotes the minimum amount of the same attributes value between HP_{CAR} and HP_{LR}.

Input: a **LR**
Output: a suitable **CAR**

Step 1: If the **CADT** is not Empty,
Then use the HP in LR 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 $(\mathrm{LO}_{ij} \in \mathbf{CAR}_{set}) \neq (\mathrm{LO}_{mn} \in \mathbf{LR})$.
Step 4: Delete these CARs from **CAR**$_{set}$, if $|(\mathbf{B} \in \mathbf{CAR}_{set}) - (\mathbf{B} \in \mathbf{LR})| \geqq \alpha \times \mathbf{B} \in \mathbf{LR}$.
Step 5: Delete these CARs from **CAR**$_{set}$, if the number of HP attributes with similar value in CAR compared with LR < $\mathbf{S}_{Min}$.
Step 6: Delete these CARs from **CAR**$_{set}$, and if the similarity between CAR in **CAR**$_{set}$ and LR according to the $\mathbf{Similarity}_{LP}(\mathrm{LP}_{LR}, \mathrm{LP}_{CAR}) < \gamma$.
Step 7: If $\exists$ a CAR$\in$ **CAR**$_{set}$ whose attribute values in HP and LP is the same as LR, **Then** goto **Step 9**.
Step 8: do the Content Adaptation Process (CAP) according to the LP in **LR** 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**$_{new}$.

Example 6 Based on the data in Table 1, given a new Learner Request (LR), LR = (LO_{ij}, (B, HP, LP)) = (LO, (90 KB, ⟨**2**, 528, 384, 320, 480, **24**, 32, 120⟩, ⟨5,JGBP,1,1 ⟩)) and a new CAR 16 = (LO_{ik}, (150 KB, ⟨1, 133, 128, 480, 640, 16, 16, 44⟩, ⟨12, GJBP, 0, 1⟩)) in the CAR Pool, according to the CADT in Fig. 3 and Adaptation Decision Process Algorithm (ADPAlgo), we can find the rule: *if Machine*

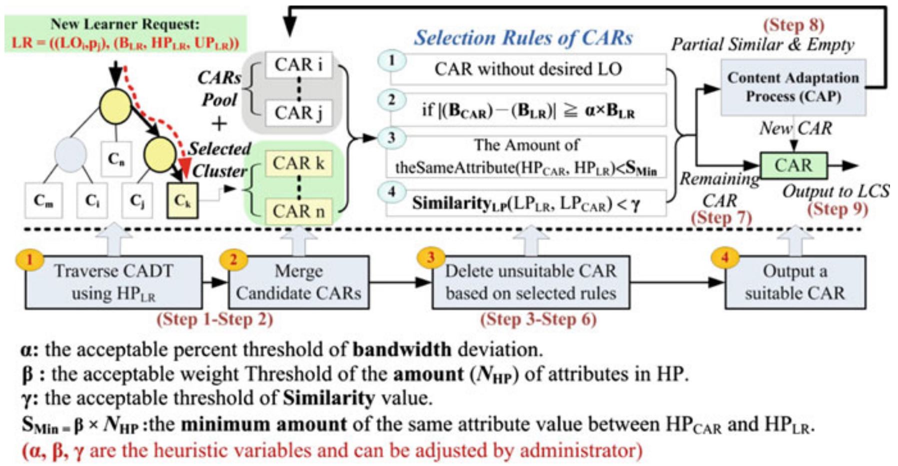


Fig. 5 ADP process

Type (MT) = '2' *and Color Depth* (CD) = '24,' *then* '**C_2**,' so that we can use the CAR ID, {2, 5, 7, 9, 11, 12, 13, 15}, of C_2 in Table 2 and CAR 16 in CAR Pool to select a suitable CAR (**Steps 1 and 2**). Then, CAR 16 is deleted due to ($LO_{ik} \neq LO_{ij}$), and CARs 2, 5, 7, 11, 12, and 13 are deleted due to their bandwidth deviation $\geqq 45(\boldsymbol{\alpha} \times \mathbf{B})$ while $\boldsymbol{\alpha}$ is 0.5 and **B** is 90 KB (**Step 3** through **Step 4**). Afterward, CAR 15 with 7 similar attributes while $\mathbf{S}_{min} = 0.9(\beta) \times 8(N_{HP}) = 7.2$ and the similarity value = 0.772 ($>\gamma$=0.6) compared with an LR is a suitable CAR for the user (**Step 5** through **Step 6**).

However, because CAR 15 is not completely the same as the LR, a new CAR, CAR 17, will be created by the Content Adaptation Process (CAP) based on the LR and stored in the CAR Pool for the next similar learner request (**Step 8**). Thus, the CAR Pool will hold two new CARs, i.e., {16 and 17} and the adapted content version based on CAR 15 will be delivered to a learner directly. Because the content version of CAR 15 was adapted according to the previous similar learner request, the CAP process does not need to be executed again. Therefore, the adaptation and transcoding latency can be omitted and saved.

5.3 Learning content synthesizer

The Learning Content Synthesizer (LCS) aims to compose appropriate personalized learning content according to the adaptation decision of the PLCAM based on diverse learner preferences. As stated previously, when dealing with a new LR without any suitable existing adapted content to be delivered, the CAP will decide a corresponding MP_{set} to transcode the associated media resources. Hence, given that a page has *n* media resources and its corresponding $MP_{set} = \{MP_1, MP_2, \ldots, MP_m\}$, where $1 \leqq m \leqq n$, it is implied that the (n–m) resources do not need to be transcoded and shown due to the satisfaction degree and Switch Attribute (SA) of media, e.g., PS and AS in the LP. To notify users, media resources that are not shown will be automatically replaced by some additional annotations from the SCORM metadata. To efficiently manipulate

Algorithm 4

Algorithm 4: Learning Content Synthesis Algorithm (LCSAlgo)
Symbol Definition:
$\mathbf{T}_{CAP}$: the spending time of executing the Content Adaptation Process (CAP).
$\mathbf{T}_{ADP}$:the spending time of executing the Adaptation Decision Process (ADP).
$\mathbf{T}_{deliver}$: the estimated deliver time of the media version (V) in $\mathrm{MP}_{i.}$
$\mathbf{T}_{score}$: the minimum threshold of Satisfaction for the content adaptation process.
$\mathbf{T}_{used}$:The used time of MPs, which needn't be re-adapted.
DT: the maximum available delivery time, DT $\in$ LP in LR.
MO: denotes a Media Object $\in$ LO.
$\mathbf{MP}_{set} = \{\mathrm{MP}_1, \mathrm{MP}_2, \ldots, \mathrm{MP}_k\}$: stores all appropriated MPs selected by CAPAlgo.
$\mathbf{Media}_{adapt}$: store the media objects, which need to do content adaptation process
r_γ: denotes the original media resource in a page.
Tr_γ: denotes the transcoded media resource.

Input: a **LR** with corresponding $\mathbf{MP}_{set}$ and $\mathbf{LO}_{ij}$.
Output: an adapted and transcoded learning content version, XHTML.

Step 1: if (CAR = ADPAlgo(LR)) = null**,**
Then
(1.1) Estimate the T_{CAP} and T_{ADP}
(1.2) MP_{set} = CAPAlgo(LR, DT$-$(T_{CAP} + T_{ADP})), where $\mathbf{T}_{MDT}$ in CAPAlgo = (DT$-$(T_{CAP} + T_{ADP})).
Else
(1.3) for each $\mathrm{MP}_i \in \mathrm{MP}_{set}$ in CAR
(a) Calculate $\mathbf{Sat}_i = Satisfaction_{QualityOfMedia}(\mathrm{LR}, \mathrm{MP}_i)$
(b) Calculate $\mathbf{T}_{deliver}$ = size (S)$\in$ MP_i/ Bandwidth(B)$\in$LR
(c) Calculate$\mathbf{T}_{transcoding} =$
$$\begin{cases} 0, \textit{ if physical file of } \mathrm{MP}_i \textit{ is in Learning Object Repository } (LOR) \\ \textit{Estimate transcoding time from nearest physical file in LOR} \end{cases}$$
(d) Calculate Satisfaction Score of $\mathrm{MP}_i = SatisfactionScore_{MP}(\mathrm{Sat}_i, DT/(the\ number\ of\ \mathrm{MP}_i \in \mathrm{MP}_{set}), \mathrm{T}_{deliver} + \mathrm{T}_{transcoding})$
(e) If Satisfaction Score of $\mathrm{MP}_i < \mathrm{T}_{score}$, then add media object (MO) of MP_i into $\mathbf{Media}_{adapt}$
(1.4) Estimate the T_{CAP} and T_{ADP}
(1.5) for each $\mathrm{MP}_i \in \mathrm{MP}_{set}$ and $\notin$ $\mathbf{Media}_{adapt}$**,**
(a) Calculate $\mathrm{T}_{used} = \mathbf{T}_{deliver+}\mathbf{T}_{transcoding}$
(1.6) for each $\mathrm{MO}_k \in \mathbf{Media}_{adapt}$**,**
(a) the MP_i of MO_k = CAPAlgo(LR, DT$-$(T_{CAP} + T_{ADP} + T_{used})), where $\mathbf{Media}_{req}$ in CAPAlgo = $\mathbf{Media}_{adapt}$ and $\mathbf{T}_{MDT}$ in CAPAlgo = (DT$-$ (T_{CAP} + T_{ADP} + T_{used})).
(b) replace the original MP of $\mathrm{MO}_k \in \mathrm{MP}_{set}$ of CAR by MP_i.
Step 2: **for** each media resource, $r_{\gamma,}$ in a page $\mathbf{p}_i$**.**
(1) apply $\mathbf{MP}_\gamma \in \mathbf{MP}_{set}$, to transcode the r_γ into the tr_γ.
Step 3: transform the original HTML into XHTML format
Step 4: replace all r_γ by tr_γ into the XHTML**.**
Step 5: replace all unshown media resources by the useful annotation from SCORM metadata.
Step 6: output the XHTML with associated transcoded media resources.

the diverse versions of learning content, a page's original HTML will be transformed into a well-formed XHTML and tree-like Document Object Model (DOM) structure (2010). The details of the LCS are described in Algorithm 4.

Example 7 Assume the CADT structure has already been built. Then, given LR = ((LO$_2$, p$_1$), (500 KB, ⟨1, 1000, 448, 480, 800, 24, 16, 140⟩, ⟨2, GPJB, 1, 0⟩)) and the minimum threshold of satisfaction, $T_{score} = 0.7$. After the ADPAlgo(LR) process, there are two corresponding MOs in MP_{set}, i.e., $MP_{set} = \{MP_1, MP_2\} = \{$("M$_{1v0}$", image, ⟨800, 400, 16, J⟩, 160), ("M$_{2v1}$", image, ⟨480, 700, 24, P⟩, 50)$\}$. Thus, assume $\mathbf{T}_{ADP} = 1$ s, and $\mathbf{T}_{deliver}$ of MP$_1$ = (size in MP$_1$)/ (bandwidth in LR) = 160 KB/ 500 KB = 0.32 s and $\mathbf{T}_{deliver}$ of MP$_2$ = 50 KB/ 500 KB = 0.1 s **(Step 1.3.b)**. The associated Media Object, MO$_2$, of MP$_2$ does not need to be adapted again because MO$_2$ has the adapted media version ($\mathbf{V}_1$) based on MP$_2$, i.e., M$_{2v1}$, in the LOR. On the contrary, M$_{1v0}$, which is the original version (V$_0$) of MO$_1$, must be adapted according to the definition of MP$_1$ before it can be delivered to the learner. Therefore, assume $\mathbf{T}_{transcoding} = 1$ s to adapt M$_{1v0}$ into the nearest version, i.e., ("M$_{1v2,}$" image, ⟨800, 400, 16, B⟩, 500)**(Step 1.3.c)**.

In addition, according to the result of Example 4, the **Sat**$_1$ of MP$_1$ is 0.463 by the equation: $Satisfaction_{QualityOfMedia}$(LR,MP$_1$) **(Step 1.3.a)**. Consequently, the satisfaction score of MP$_1$ is calculated by:

$$SatisfactionScore_{MP}(0.463, 2/\,2, 0.32\ +\ 1) = \frac{0.463}{1 + (1.32 - 1)/1} = 0.35.$$

Besides, **Sat**$_2$ is calculated by:

$$\begin{aligned}&Satisfaction_{QualityOfMedia}\,(\text{LR},\ \text{MP}_2) = 0.5 \times SimofNum(480 \in \text{HP},\ 480)\\&\quad + 0.1 \times SimofNum(800 \in \text{HP}, 700) + 0.3 \times SimofNum(24 \in \text{HP},\ 24) + 0.1\\&\quad \times SimofImage_{Type}(\text{“GPJB”} \in \text{LP, P}) = 0.9625\ (\mathbf{Step1.3.a}).\end{aligned}$$

Consequently, the satisfaction scores of MP$_2$ is calculated by:

$$SatisfactionScore_{MP}(0.9625,\ 2\ /\ 2, 0.1\ + 0) = \frac{0.9625}{1 + (0.1 - 1)/1} = 0.9625,$$

because the $T_{transcoding} = 0$ s. for adapting M$_{2\text{v}1}$ (**Step1.3.d**).

Because the satisfaction score of MP$_1$ = 0.35, which is less than 0.7 (T_{score}), and the satisfaction score of MP$_2$ = 0.9625, which is larger than 0.7, the MO of MP$_1$, M_{1v0}, will be added to the $Media_{adapt}$ for further adaptation process, i.e., $\mathbf{Media_{adapt}} =$ {"M$_{1\text{v}0}$"} (**Step1.3.e**).

Assume $\mathbf{T}_{CAP} = 0.2$ s and $\mathbf{T}_{ADP} = 1$ s for the "M$_{1v0}$" **(Step 1.4)**, "M$_{2v1}$" of MP$_2$ doesn't need to be adapted again, so the $\mathbf{T}_{used}$ for MP$_2$ = $T_{deliver}$ + $T_{transcoding}$ = 0.1 + 0 = 0.1 **(Step 1.5).** Afterward, we can get an MP$_3$ = ("M$_{1v0,}$" image, ⟨480, 240, 24, P⟩, 20) for the "M_{1v0}" by calling the CAPAlgo(LR, T_{MDT}) = CAPAlgo(LR, DT−(T_{CAP} + T_{ADP} + T_{save})) = CAPAlgo(LR, 2−(0.2 + 1 + 0.1)), where $\mathbf{Media}_{req}$ in CAPAlgo = $\mathbf{Media}_{adapt}$ = {MO_1} **(Step 1.6.a)**. Therefore, MP_{set} = {MP_3, MP_2} = {("M_{1v0}," image, ⟨480, 240, 24, P⟩, 20), ("M_{2v1}," image, ⟨480, 700, 24, P⟩, 50)} **(Step 1.6.b)**. According to the MP_{set}, the media version, M_{1v0}, of MP_3 will be transcoded by the definition of MP_3 first, and then output the XHTML with associated transcoded media resources (**Step 2** through **Step 6**).

6 Implementation and experiment

The prototypical PLCAM system was developed on an Apache Server, PHP, Perl and C Language, and a SCORM-compliant LOR. Implementation details and experimental results are described in this section.

6.1 Mobile learning scenario for the personalized learning content adaptation mechanism system

The mobile learning scenario for the prototypical PLCAM system deploys a SCORM-compliant Learning Object Repository (LOR) that allows teachers to upload teaching materials. Students, in turn, can search and download the desired learning content onto their mobile devices. Students can log in to the system through their mobile devices and manually configure their own Leaner Preference (LP) with Weight Vector (WV). They may then select the learning content to download, and proper personalized learning content will be adapted and delivered by the PLCAM according to the students' Hardware Profile (HP) and LP, and the status of the wireless networks. As the number of Learner Requests (LRs) increases, the Learning Content Adaptation Management Scheme (LCAMS) will automatically rebuild the Content Adaptation Decision Tree (CADT) and manage the adapted version of the Media Object (MO) in the LOR.

The operational flow of the PLCAM is illustrated in Fig. 6. First, a student uses an ID to log in to the prototypical PLCAM system [**Step (a)**]. After logging in, a student can use the menu on the index page of the LOR to manually configure the following: (1) the HP setting, (2) the LP setting, (3) browse the content of the LOR, and (4) read the system manual [**Step (b)**]. The PLCAM can automatically detect the **HP** of a mobile device and the current bandwidth of a wireless network, as shown in the "Dynamic Attributes" [**Step (c)**]. Next, a student can use the "User Preference Configuration" to manually define the data of an **LP**, such as the preferred maximum Delivery Time (DT), presentation ratio of audio and picture, and the preferred priority of picture, etc. [**Step (d)**]. After configuring the HP and LP, a student can click "(3) browse the content of LOR" in **Step (b)** to browse the contents by the "Category Page" of the LOR [**Step (e)**]. The system will list the learning content stored in the selected categories [**Step (f)**]. A student can also select interesting learning content to browse its "SCORM metadata" and "Table of Contents" [**Step (g)**]. Consequently, the PLCAM will adapt the chosen content according to the mobile device profile (HP),

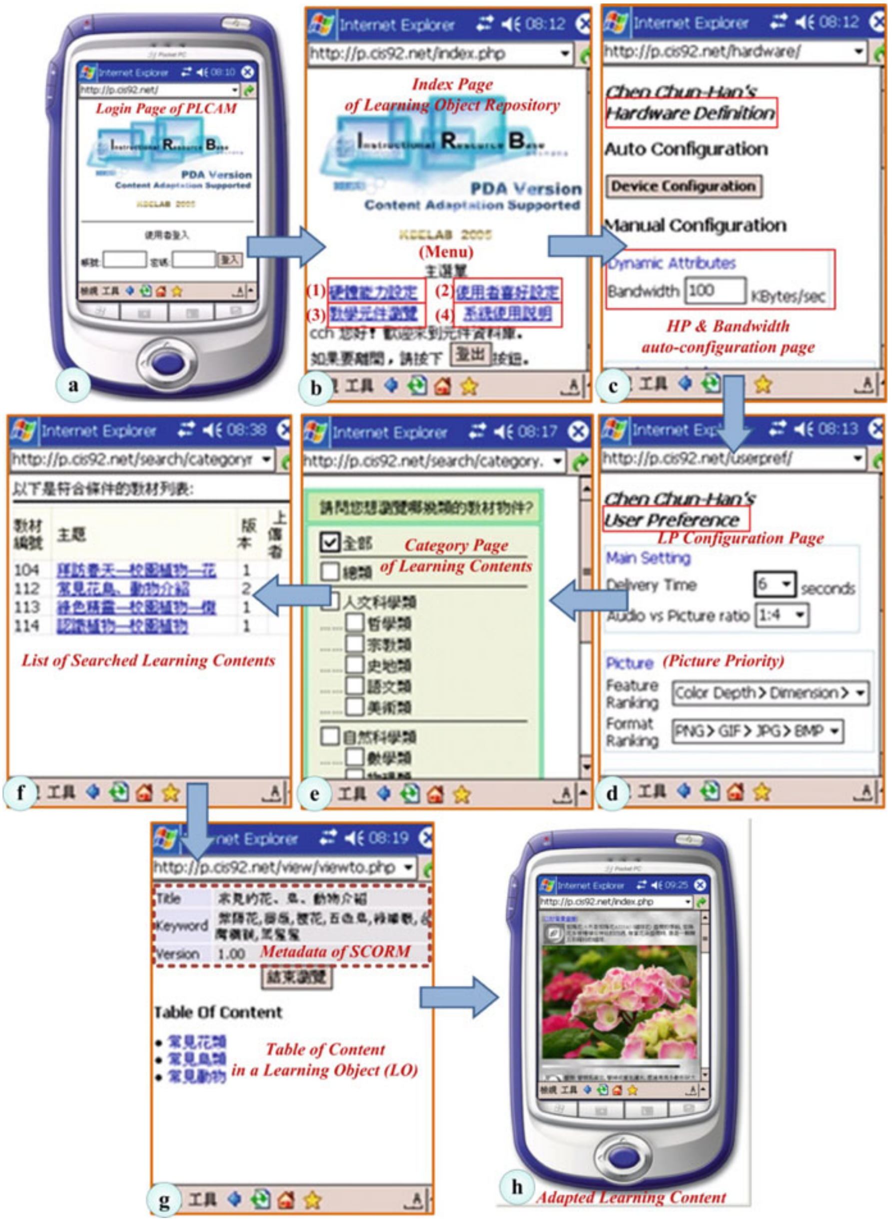


Fig. 6 Operational flow for a user to retrieve the learning content by the PLCAM system

LP, and the current Bandwidth (B) together, and offer personalized learning content to the student's mobile device [**Step (h)**].

Figure 7 illustrates screenshots of the PLCAM delivering proper personalized adapted learning object content to the similar LR without waiting for the

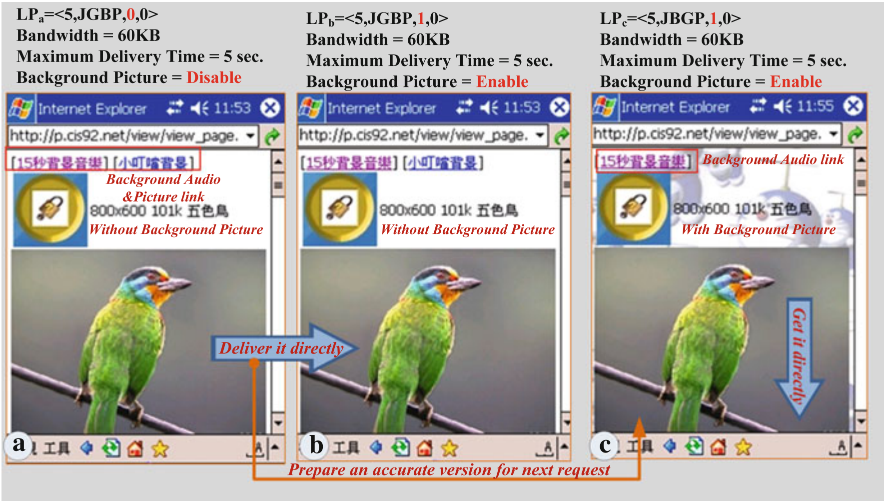


Fig. 7 **a** Adapted content version of LP_a; **b** delivered the adapted content version of LP_a for LP_b due to the higher similarity; and **c** delivered the content version created by LP_b in advance for LP_c

adaptation time of the requested content. The performance and effectiveness of the PLCAM will be evaluated and described in Sect. 6.2. Assume there is an existing adapted content version created by $\mathrm{LP}_a = \langle 5,\mathrm{JGBP},0,0\rangle$ of learner A, as shown in Fig. 7a. This existing version will be selected in advance and delivered directly to the new request with $\mathrm{LP}_b = \langle 5,\mathrm{JGBP},1,0\rangle$ of learner B, due to the higher similarity estimation, as shown in Fig. 7b. Therefore, learner B does not need to wait for the adaptation to take place again. In the meantime, the PLCAM will prepare an accurate content version with a background picture to meet $\mathrm{LP}_b = \langle 5,\mathrm{JGBP},1,0\rangle$, which is stored in the LOR for the next similar request. For example, this prepared version of LP_b can be delivered directly to meet the new $\mathrm{LP}_c = \langle 5,\mathrm{JBGP},1,0\rangle$, as shown in Fig. 7c.

Figure 8 shows several experimental screenshots of the PLCAM system executed on a PDA according to diverse user needs. Figures 8a and b illustrate adapted content based on the same HP and LP with different adaptation parameters under different bandwidth values, respectively. As stated in Example 1, the attributes of the LP and HP can be extended to meet the various requirements. Thus, a new attribute in the LP, called *Preferred Picture Property Ordering* (PPPO), includes three properties: Dimension (D), Color Depth (C), and Quality (Q). This attribute is used to define the learner's preferred order of image properties. For instance, like the attribute PPFO, a string, DQC, denotes that the order of image priorities is $D > C > Q$. Hence, we added the PPPO attribute into the LP and changed several parameters, e.g., Delivery Time (DT) and Audio Switch (AS), to test the results of the learning content adaptation process. As shown in Fig. 8c, according to the new LP setting and original HP, the property of the picture was changed and the audio, background picture, and icon were replaced by hyperlinks with annotation text. We further evaluated the learning content adaptation capability by changing the screen's horizontal size and color depth of the

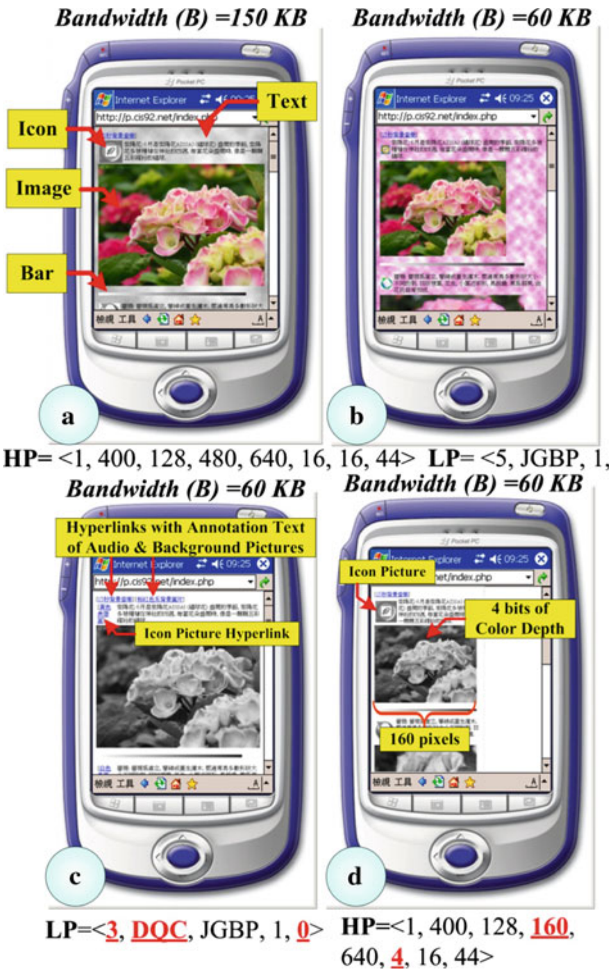


Fig. 8 Screenshots of the learning content adaptation process performed by the PLCAM system

HP while using the same LP in Fig. 8c. The desired adapted learning content was delivered by the PLCAM system, as shown in Fig. 8d.

The PLCAM system also includes a monitoring interface of the LCAMS Web server, used to monitor the latest system status and maintain the CADT. As shown in Fig. 9, the "Assign Cluster Label" function button can be used to perform the LP Clustering Algorithm (LPCALgo) for grouping the historical CARs into several clusters according to the learners' LPs, where the resultant clustered information of the LPCALgo will be shown in the bottom-left part of Fig. 9. Furthermore, the CADT can be reconstructed by the "Rebuild Decision Tree" function button. Its graphical presentation and rule-based representation will be automatically shown in the top-left part of Fig. 9. The right-hand side of Fig. 9 will list all of the CARs.

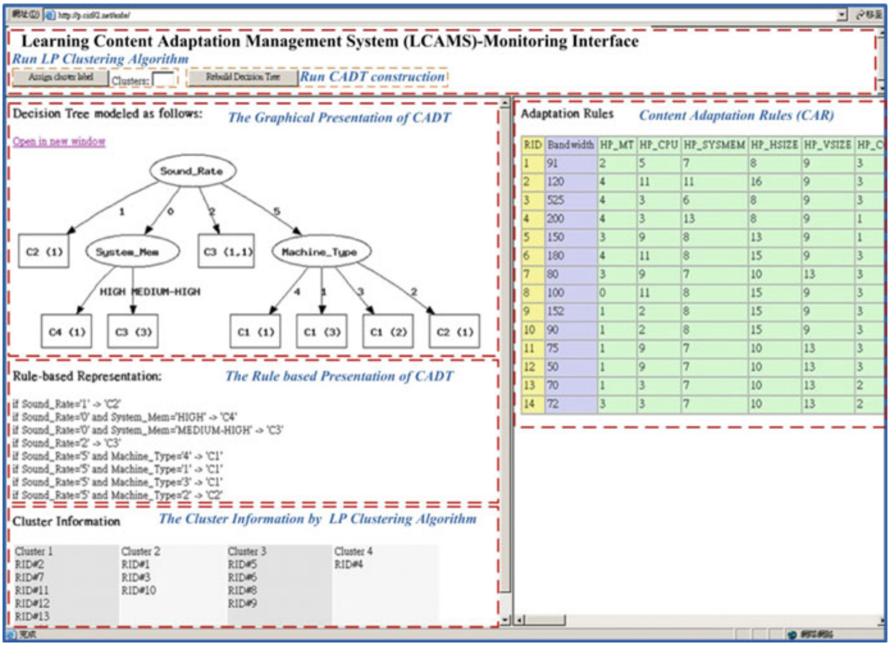


Fig. 9 Monitoring interface screen of the LCAMS Web server in the PLCAM system

6.2 The experimental results

To evaluate the PLCAM system, actual experiments and simulated experiments were performed. The details and results are described in the following subsections.

6.2.1 Results of actual experiments

In the actual experiments, performance of the prototypical PLCAM system was evaluated by the experimenters in terms of the personalized learning content delivering process, including the Content Adaptation Process (CAP), the Adaptation Decision Process (ADP), and the Learning Content Synthesizer (LCS), and the dynamic bandwidth detection scheme, based on the SCORM-compliant learning content in relation to "the plants in campus" shown in Sect. 6.1. These characteristics were tested to observe and evaluate the resultant Delivery Time (DT) and the transmission data size according to various bandwidth settings for different requests. The performance of the PLCAM in terms of DT compared with inadaptation and static adaptation approaches was evaluated as well.

As mentioned in Sects. 1 and 2, the DT plays an important role in affecting one's learning performance in mobile learning environments. Therefore, a bandwidth detection scheme was developed to automatically detect the latest network bandwidth for providing the learner with more precise personalized learning content with higher fidelity. As shown in Fig. 10a, with the decrease of "actual" network bandwidth, the

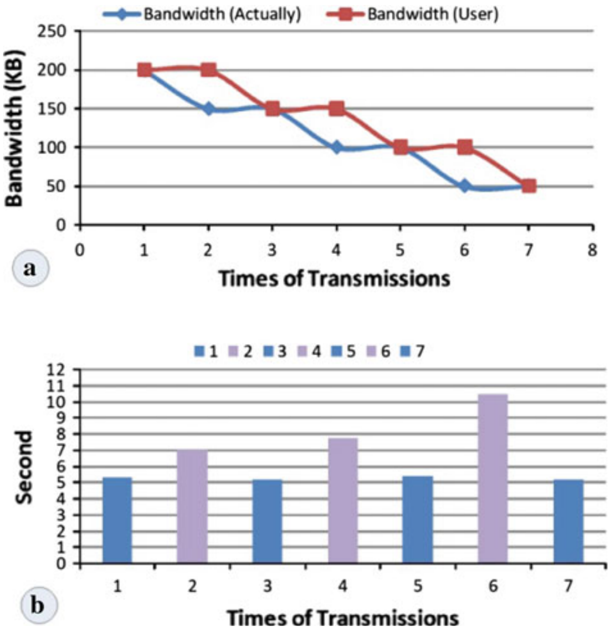


Fig. 10 Experiment results of the automatic dynamic bandwidth detection scheme

bandwidth that a user can consume will decrease as well by means of monitoring each transmission time compared with the setting of the user's desired maximum DT. For example, in Fig. 10b, a user's functional bandwidth has been updated at 3, 5, and 7 (times) due to the detection of the long delivery latency at 2, 4, and 6 (times) (maximum DT = 5 s).

Figure 7 shows the effectiveness of the PLCAM in delivering proper personalized adapted learning content that meets a similar LR without waiting for the CAP to be completed. Therefore, Fig. 11 illustrates the DT, which consists of the data transmission time and the PLCAM adaptation time, in terms of the different requests and the size of transmission data based on the various bandwidth settings. In Fig. 11a, assume the definition of the maximum DT is 5 s, during the first request for a learning object, the average DT is about 8.416 s, including the content adaptation process (about 3 s) and the actual content delivery. On the contrary, the average DT during the second request can be controlled around 5.238 s to meet the constraint of the maximum DT without repeating the content adaptation process. Figure 11b shows that the size and quality of transmission data can be increased gradually with the increase of usable bandwidth by the aforementioned dynamic bandwidth detection scheme based on the definition of maximum DT.

Assume that there is a learning object, which contains a Waveform Audio Format (WAV) file with 660,768 bytes and six pictures with 1,016,392 bytes. The original size is about 1.8 Mbytes. The learner specified the maximum tolerable DT to be 5 s We observed the transmission results based on the various bandwidth settings in terms of the approaches, which included the inadaptation, static adaptation, and PLCAM. In

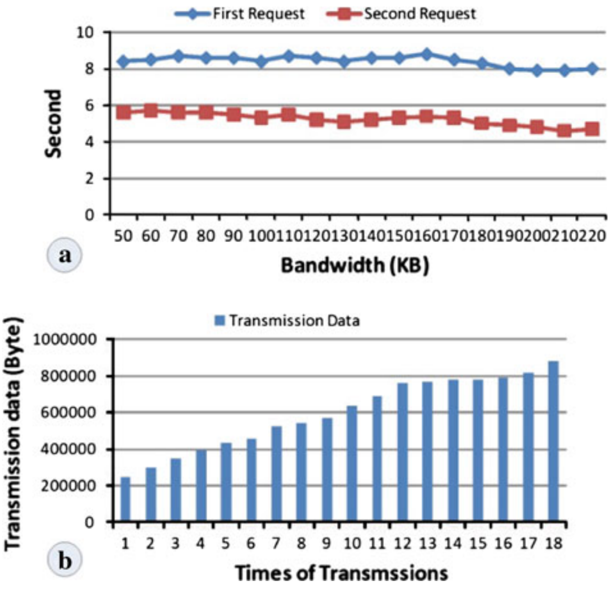


Fig. 11 DT of different requests and transmission data size based on various bandwidth settings

Fig. 12, the inadaptation approach transmits content without employing the content adaptation process and spends much more DT than the static adaptation approach and the PLCAM. In this example, the static adaptation approach prepared three versions of learning content in advance, i.e., 200, 170, and 140 KB. Therefore, within the bandwidth range from 140 to 220 KB, the DT is almost the same between the static adaptation approach and the PLCAM. However, the static adaptation approach cannot consistently provide users with the appropriate content version according to the various bandwidths; thus, this approach spent a great deal of time gradually decreasing the bandwidth from 140 to 50 KB. On the contrary, the PLCAM is still able to offer a stable delivery time and the proper personalized adapted content to meet the diverse user needs.

6.2.2 Results of simulated experiments

To evaluate the performance and effectiveness of the PLCAM in depth, several simulated experiments were carried out, emulating a large number of diverse user requests with Learner Preferences (LPs) and Hardware Profiles (HPs) to access the desired Learning Objects (LOs) from the Learning Object Repository (LOR). The perFORmance and satisfaction degree of the PLCAM in terms of: (1) Learning Content Adaptation Management Scheme (LCAMS) with Content Adaptation Decision Tree (CADT) in Figs. 13, 14, 15, and 16; (2) the parameter setting of the LP clustering algorithm in Fig. 17; and (3) the CADT maintenance process in Figs. 18 and 19 were evaluated based on the different experimental conditions, including bandwidth, LPs, and devices.

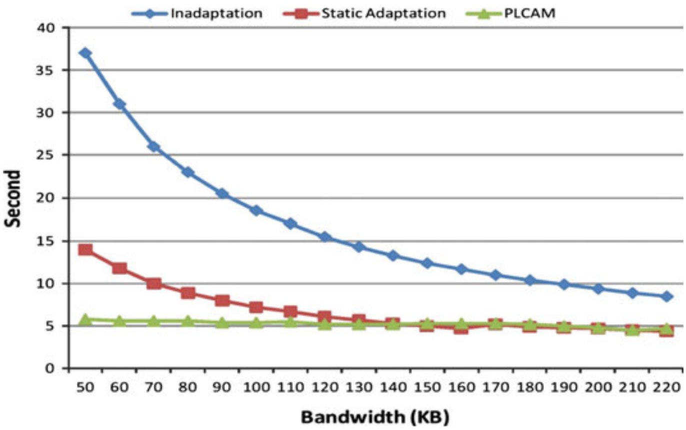
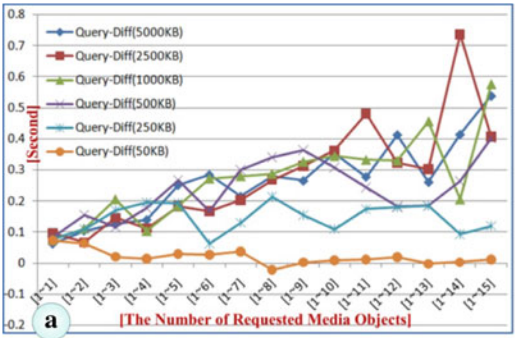
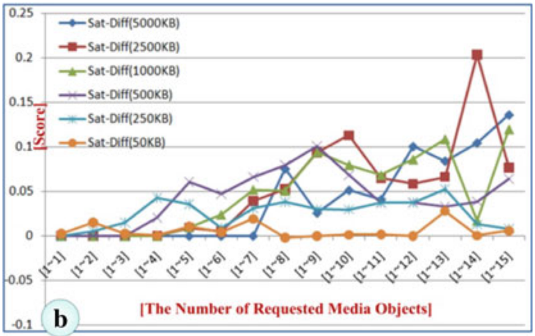


Fig. 12 Comparison among the inadaptation, static adaptation, and PLCAM approaches

Fig. 13 Comparison of **a** the difference of query time; and **b** the difference of satisfaction between the PLCAM without and with the CADT based on different bandwidths and requested MOs

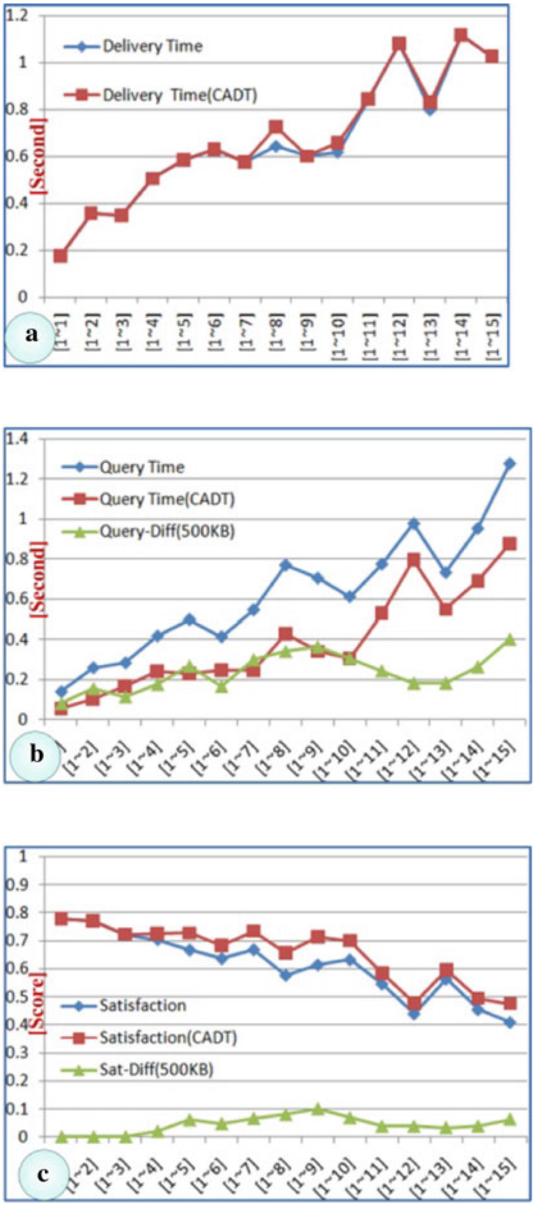




The simulated experiments were executed on a computer with a 1-GHz Central Processing Unit (CPU), 1 G of Random Access Memory (RAM), and a Windows XP Operating System (OS). Table 5 lists the HP data used to perform the following experiments.

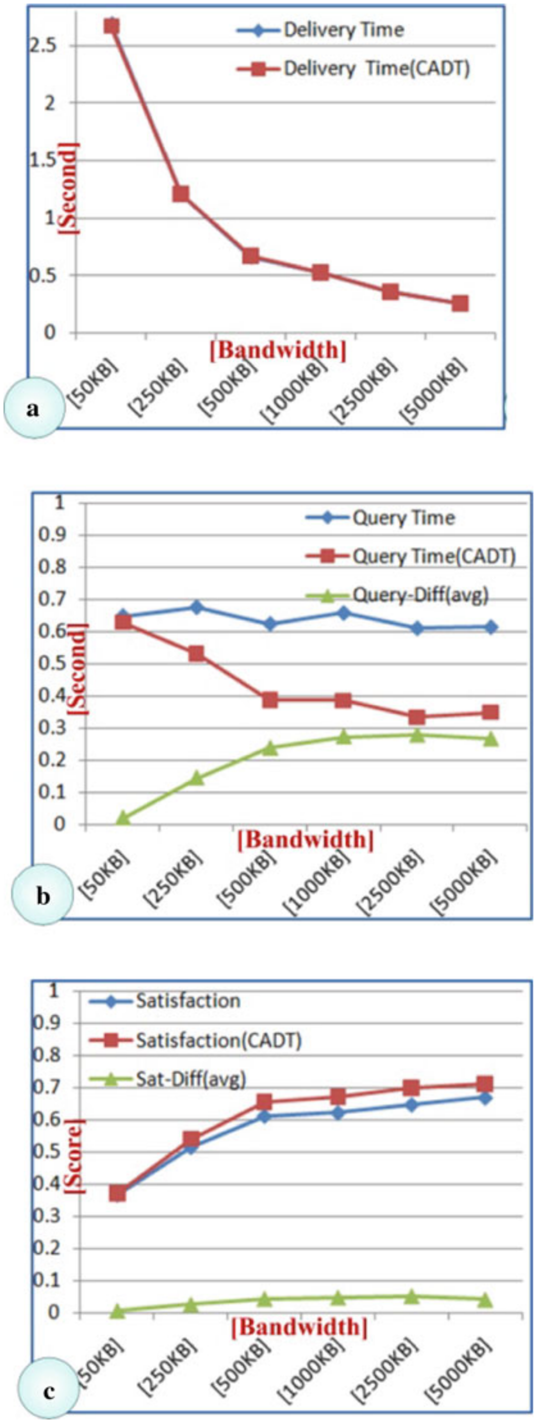
The LCAMS in the PLCAM uses the CADT structure to efficiently determine the appropriate personalized learning content to meet the diverse learner requests.

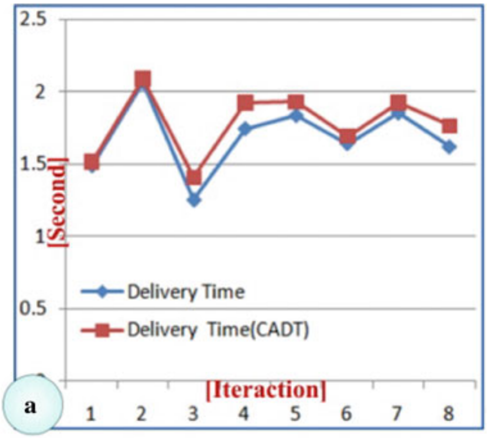
Fig. 14 Comparison of **a** the delivery time; **b** the query time; and **c** the satisfaction score between the PLCAM without and with CADT based on 500 KB bandwidth and different requested MOs

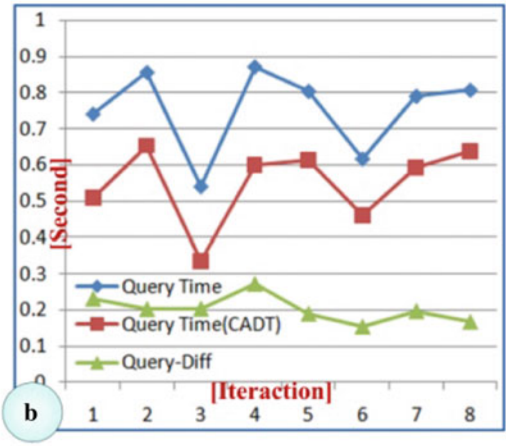


Therefore, we analyzed the performance and differences between the PLCAM without and with the CADT to perform the content adaptation based on different bandwidths, 5000, 2000, 1000, 500, 250, 50 KB, and the number of requested Media Objects (MOs) from 1 to 15, i.e., [1–15]. During this simulated experiment, each of the 250 Learner Requests (LRs) was generated by LP = (Maximum Delivery Time = 1 s, JPBG, 1/0,

Fig. 15 Comparison of **a** the average delivery time; **b** the average query time; and **c** the average satisfaction score between the PLCAM without and with the CADT based on different bandwidths and requested MOs







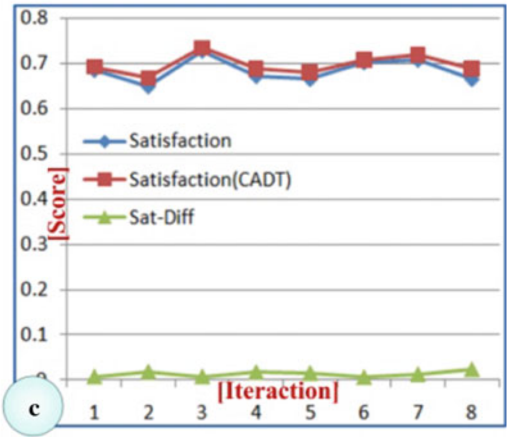


Fig. 16 Comparison of **a** the delivery time; **b** the query time; and **c** the satisfaction score between the PLCAM without and with the CADT on random bandwidths [50 KB, 500 KB], random maximum DT [1,8], random requested MOs [1,9], and eight HP data points in Table 5

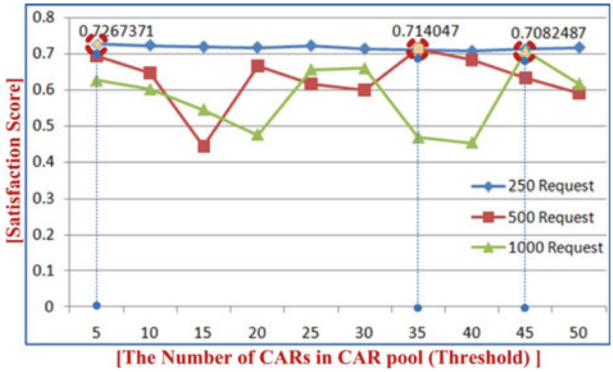


Fig. 17 Most suitable threshold of rebuilding CADT based on the different amount of CARs in the CAR pool

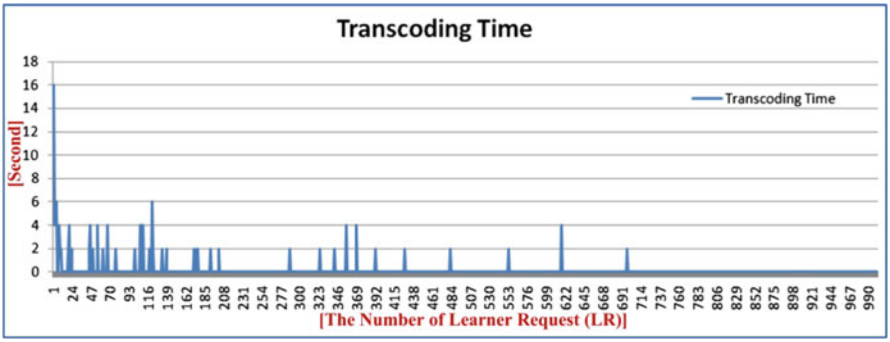


Fig. 18 Resultant transcoding time of the PLCAM with auto-adjustment scheme

1/0) and the random HP ID between 1 and 6, i.e., [1,6], in Table 5. The results of the simulated experiment are shown in Figs. 13, 14, and 15, respectively.

In Fig. 13, the Query-Diff and Sat-Diff denote the difference of query time of determining the suitable MP_{set} and the satisfaction score between the PLCAM without and with the CADT, respectively. Figure 13a shows that the Query-Diff explicitly increases with the increase of bandwidth ≧ 250 KB and the number of requested MOs ≧4, which shows that the CADT can efficiently speed up the performance of the Adaptation Decision Process (ADP).

Figure 13b indicates that the Sat-Diff also increases if the bandwidth ≧ 500 KB and the number of requested MOs ≧ 7, which shows that decrease of query time can enhance the satisfaction score because response time is an important factor in user satisfaction. As for the bandwidth = 50 KB, the Sat-Diff and Query-Diff are very low because the available DT is insufficient to determine the MP_{set} with a better satisfaction score in the ADP. On the contrary, the PLCAM without the CADT needs much more time to determine the suitable MP_{set} from the MP database while the number of requested MOs increases.

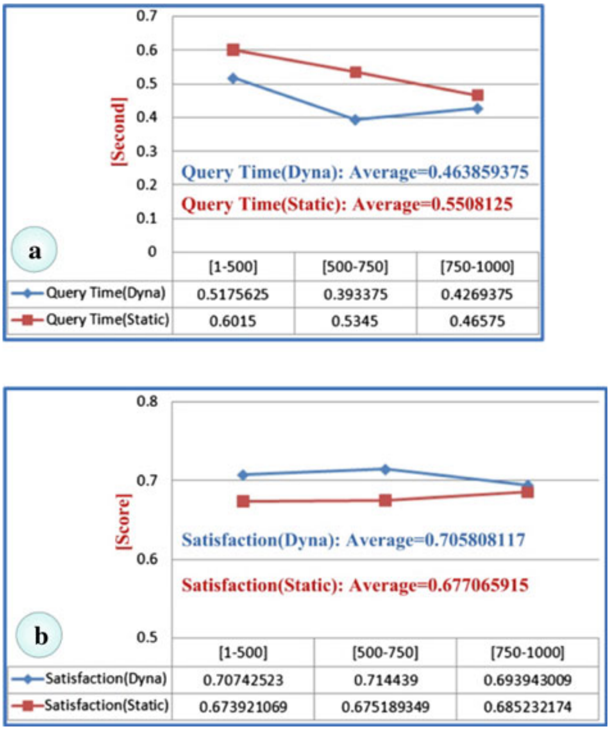


Fig. 19 Comparison of **a** query time; and **b** the satisfaction score of the PLCAM between dynamic-threshold and static-threshold by the random LRs and eight HP data points in Table 5

Figure 14 shows the delivery time, the query time, and the satisfaction score between the PLCAM without and with the CADT based on 500 KB bandwidth only. In Fig. 14a, the Delivery Time (CADT), consisting of physical data transmission time and transcoding time, is almost the same as the PLCAM without CADT approach. In Fig. 14b, the difference of query time (Query-Diff) is from 0.08 to 0.4 s, which saves 8–40% time consumption in terms of DT = 1 s Furthermore, although the PLCAM uses the CADT to improve the performance of the content adaptation process, a higher satisfaction score than the score obtained without the CADT, can be maintained, as seen in Fig. 14c.

Regarding the influence of bandwidth between query time and the satisfaction score, Fig. 15a shows that the average DT is almost the same without and with the CADT. This finding indicates that the CADT can determine similar personalized learning content like the PLCAM without CADT. Also, query time (CADT) will decrease with the increase of bandwidth, while query time without the CADT is almost the same, as seen in Fig. 15b. This is a 2–27% (average 20%) time consumption savings in terms of DT = 1 s. Therefore, the average satisfaction score (CADT) is also better than "without CADT," as presented in Fig. 15c.

To evaluate the performance of the PLCAM in actual mobile learning environments, we emulated diverse LRs actually used by the PLCAM with randomized LRs, which

Table 5 HP in LR data used for the simulation experiments

ID	HP in CAR	Machine type
1	⟨**2**, 600, 384, 320, 480, 24, 16, 40⟩	Cell phone
2	⟨**1**, 1000, 576, 480, 800, 24, 32, 100⟩	PDA
3	⟨**1**, 1000, 448, 480, 800, 24, 16, 140⟩	PDA
4	⟨**2**, 528, 288, 320, 480, 32, 8, 20⟩	Cell phone
5	⟨**2**, 528, 384, 320, 480, 16, 32, 120⟩	Cell phone
6	⟨**2**, 528, 288, 320, 480, 16, 8, 30⟩	Cell phone
7	⟨**0**, 2000, 8000, 1366, 768, 32, 32, 500⟩	Notebook
8	⟨**0**, 1200, 4000, 1366, 768, 32, 16, 300⟩	Notebook

had random maximum DT between 1 and 8 s, [1,8], the random bandwidths between 50 and 500 KB, [50 KB, 500 KB], and the random number of requested MOs between 1 and 9, [1,9]. We tested this simulated experiment for eight iterations, each of which used eight participant HP data in Table 5 and generated 250 LRs to test the PLCAM system. Figure 16 shows the results of the experiment. The Delivery Time (CADT) is a bit higher than what is seen without the CADT, as shown in Fig. 16a. The Query Time (CADT) is still better than what is observed without CADT, from about 0.15 to 0.27 s (average is 0.2 s), as seen in Fig. 16b, and the satisfaction score is almost the same and stable around 0.7 during eight iterations. These results show that the PLCAM with the CADT can achieve better and more stable performance regarding learning content adaptation and the satisfaction degree in simulated actual learning environments.

To analyze the parameter setting of the LP Clustering Algorithm (LPCALgo), we used different parameter settings to test the satisfaction score of the PLCAM system based on LP = (1, JPBG, 1/0, 1/0), HP from 1 to 6 in Table 5, bandwidth = 500 KB, and the number of MOs between 1 and 9. The parameter setting of the LPCALgo has been found as {K = 3, $T_s = 0.004$, $T_m = 2$, $T_n = 3$, $T_p = 2$}, where the PLCAM attains a better satisfaction degree.

By means of the analysis of the parameter setting of LPCALgo, we found that the number of CARs in the CAR Pool, employed to be the threshold of rebuilding the CADT, play an important role in satisfaction. Therefore, we used the aforementioned parameter setting to evaluate the performance of the PLCAM system by adjusting the threshold of rebuilding the CADT. Thus, the experimental results are shown in Fig. 17, where we find that the most suitable thresholds to rebuild the CADT are: 5 at 250 requests, 35 at 500 requests, and 45 at 1000 requests, respectively. According to these results, we can use the ordinary least squares approach to estimate the CADT rebuilding equation:

$$\textit{CADT Rebuilding Equation} : Y = 0.04857X,$$

where Y is the predicted thresholds of rebuilding the CADT and the X is the number of LRs.

For example, if the number of LRs is 750, we can use the $Y = 0.04857X = 0.04857 \times 750 = 36$ to be the threshold. Therefore, if there are 36 new Content Adaptation Rules (CARs) in the CAR Pool and the total LRs is larger than 750, the CADT maintenance process will rebuild the CADT automatically.

By means of the CADT rebuilding equation, the PLCAM can automatically maintain its CADT associated with historical CARs according to the "use situation" of learners. Therefore, in order to evaluate the performance of the PLCAM with the CADT maintenance process, we test the PLCAM by LP = ([1,8], random, 1/0, 1/0), Bandwidth = [50 KB, 500 KB], the number of media = [1,9], the HP data in Table 5 to emulate the actual use by learners.

Figure 18 illustrates time spent during the transcoding process over the course of 1000 LRs. In Fig. 18, most of the transcoding time was spent during the early phase of the requests, as opposed to the latter phase. Because the PLCAM can efficiently manage a large number of historical learners' requests and intelligently deliver proper personalized learning content with higher fidelity from the LOR to the learner directly, the transcoding time can be decreased substantially.

Figure 19 illustrates the comparison of query time and the satisfaction score of the PLCAM between the dynamic-threshold estimated by the CADT rebuilding equation and static-threshold based on the same experimental condition. According to Fig. 19, we can find that the PLCAM with dynamic-threshold can outperform the one with static-threshold.

6.3 Extensibility of the personalized learning content adaptation mechanism

In order to meet the various requirements, the PLCAM can be extended if necessary. The extensibility of the PLCAM can be described by the following two aspects.

(1) *Extensibility of the Content Adaptation Rule*

The attribute definitions of the Learner Preference (LP) and Hardware Profile (HP) can be extended to meet the various requirements. As shown in Fig. 8, we add a new attribute in the LP, called Preferred Picture Property Ordering (PPPO), to define the learner's preferred order of image properties. As mentioned in Sect. 1, the definition of LP may consist of content type preference including text, picture, audio, video, cognitive styles (including field dependence/independence (Riding and Cheema 1991)), and learning style (including Doer, Watcher, Thinker, and Feeler (Kolb 1976, 2004)). These preferences can also be extended into the LP. For example, content type preference can be represented as "TPAV" to indicate the preferred order as Text > Picture > Audio > Video; cognitive styles can be represented as "FD/FI," and the learning style can be represented as "D/W/T/F." Here, LP = ⟨7, **FI**, **D**, **TPAV**, JBGP, 1, 1⟩ represents that a learner prefers the cognitive style in field dependence, is a Doer in learning style, and identifies as "TPAV" in content type preference. The HP can also be extended by adding new attributes. For example, we can add a new hardware attribute, called networking type (including the values: Wireless LAN (WL), Global System for Mobile Communications (GSM), General Packet Radio Service (GPRS)) into the HP, and the HP can be represented as HP = ⟨3, 400, 128, 480, 640, 16, 16, 44, **WL**⟩.

Therefore, the extended definition of the CAR can be processed by the PLCAM to meet more diverse learner requirements.

(2) *Extensibility of Content Adaptation Process*

As mentioned in Sect. 5, the Content Adaptation Process (CAP) in the PLCAM can efficiently determine a corresponding MP_{set} to adapt and transcode the associated media resources of learning content. However, the different media types may be efficiently adapted and transcoded by means of different transcoding tools. Therefore, the transcoding tools or approaches in MP_{set} of the PLCAM can be extended according to the requirements of various media types. In addition, the function of Learning Content Synthesizer (LCS) in the PLCAM is used to arrange and compose the appropriate personalized learning content according to the adaptation decision of the PLCAM based solely on the diverse LPs. In theory, the LCS process can be extended or modified according to the different requirements. For example, we can adopt the existing dynamic content adaptation approach in terms of the content structure analysis stated in Sect. 2 to integrate with the LCS in the PLCAM.

7 Limitation and discussion

7.1 Limitation of the personalized learning content adaptation mechanism

We discussed the limitations of our proposed PLCAM approach in terms of the following three aspects.

(1) *Capability of the Content Adaptation Process* (CAP)

Currently, the PLCAM system can perform the CAP for the whole Media Object (MO) of learning content only, such as resizing and reformatting. It is not yet able to generate the adapted content with semantic preservation (Yang et al. 2007b; Chen et al. 2010), contextual summarization (Buyukkokten et al. 2001, 2002), etc. Although the PLCAM can theoretically be integrated with existing dynamic content adaptation approaches to enhance its content adaptation process capabilities, some unexpected issues may first need to be resolved.

(2) *Capability of the Content Adaptation Rule* (CAR)

According to the CAR definition, this rule can be represented and extended by diverse requirements, as mentioned in Sect. 6.3. However, the PLCAM was evaluated by the CAR with 4-tuple Learner Preference (LP) only. Therefore, the performance of the PLCAM with the extended CAR definition may have unexpected results.

(3) *Capability of the Similarity Measure in Clustering Process*

To apply the ISODATA clustering approach, a similarity measure must be designed to estimate the similarity value between two CARs based on the LP definition. However, in the PLCAM, the similarity measure must be defined according to the characteristics of attributes in the LP, so the variety of attributes will define how difficult efficiency in the similarity measure will be to achieve.

7.2 Discussion of the personalized learning content adaptation mechanism

The results of the actual and simulated experiments show that the PLCAM can provide learners with personalized learning content according to their diverse LRs, with a better satisfaction degree in most experimental conditions. For example, Fig. 15b illustrates that the PLCAM with the CADT can achieve up to a 20% savings in time consumption on average in terms of DT = 1 s, compared with the PLCAM without the CADT. This result indicates that the CADT can enhance the performance of Adaptation Decision Process (ADP). However, the satisfaction degree will decrease when maximum Delivery Time (DT) and bandwidth are insufficient, such as DT = 1 s and bandwidth = 50 KB. Because the ADP cannot determine the appropriate media set with high satisfaction in a short period of time, the query time of the PLCAM will increase with insufficient bandwidth and limited DT. In the meantime, the CAP will be executed several times to search the MO for a higher degree of satisfaction. In addition, the satisfaction score's definition takes the entire process time of the PLCAM into account, i.e., the response time of the system for improving the user's experience (Muntean 2008). The query of saving time cannot increase the satisfaction score explicitly because the ratio of query time to delivery time is small.

According to the classification of a content adaptation system defined in Fudzee and Abawajy (2008)), and based on comparison methods used in Nimmagadda et al. (2010), the methods employed as the criteria in this paper can be defined. These definitions can be based on what "media format" is to be adapted to support what purposes ("general" or "learning environment"), and how to adapt content ("layout generation" and "semantic preservation") to what context ("device," "network," or "user preference"). Furthermore, the maintenance, extensibility, and satisfaction of proposed systems are also primary concerns. Therefore, Table 6 shows the comparison of our approach with existing articles in relation to content adaptation in terms of the aforementioned criteria.

The differences between our proposed approach and the existing studies are: (1) we define the representation of the Learner Preference (LP) to efficiently describe the diverse requirements (i.e., device features, user preferences, and network conditions) in the mobile learning environments; (2) we define the satisfaction indicator to evaluate the quality of content adaptation; (3) we propose the automatic maintenance scheme to manage the learning content adaptation process efficiently; and (4) we propose the extension scheme of LP and the functionality of the PLCAM.

8 Conclusions

To consider the mobile devices' capabilities, wireless network conditions, and diverse learner needs together, we proposed a Personalized Learning Content Adaptation Mechanism (PLCAM). The PLCAM can efficiently manage a large number of historical learners' requests and intelligently deliver proper personalized learning content with higher fidelity directly to learners. After delivery, the PLCAM prepares an adapted content version for the next similar request. The mechanism defines an adaptation data format, consisting of learner preference, hardware profile, network

Table 6 Comparison with existing approaches. (O: Yes, ×: NO, Δ: partial, ? unknown)

Method	Hsiao et al. (2008)	Mohomed et al. (2006a,b, 2007); Chen et al. (2010)	Yang et al. (2007b)	Nimmagadda et al. (2010)	Muntean (2008)	Tong et al. (2006)	Franzoni et al. (2008)	Our approach
Media format	Web pages	Image	Web pages	Text, image, video	Web pages	Text, image, video	?	Text, image, audio, video
Automatic layout generation	×	×	O	O	×	×	×	O
Semantic preservation	×	×	O	×	×	×	×	×
Automatic maintenance scheme	O	×	?	×	Δ	×	×	O
Device features	O	O	O	O	O	O	×	O
Network condition	×	Δ	×	O	O	×	×	O
User preference	×	Δ	×	O	O	O	Learning style	O
Satisfaction evaluation	×	×	×	Screen utilization	O	×	×	O
Extensibility	O	×	?	Δ	×	×	×	O
Learning environment	×	×	×	×	O	O	O	O

conditions, and media parameters, to describe diverse learner needs, and it uses the Content
Adaptation Decision Tree (CADT) created by data mining techniques to efficiently determine the appropriate adapted contents from a Learning Object Repository (LOR).

Proper personalized learning content, which features learning resources according to a set of diverse learner preferences, can be generated and delivered to learners to improve both performance of the content adaptation process and overall user satisfaction. Additionally, the experimental results of the prototypical PLCAM system show that our proposed Learning Content Adaptation Management Scheme (LCAMS) can efficiently determine the appropriate personalized learning content that can meet diverse learner requests through the use of the CADT. The query time of this process can reap a significant 40% consumption savings. Furthermore, the PLCAM can automatically maintain the CADT associated with historical Content Adaptation Rules (CARs) according to the use of learners by means of the CADT maintenance process. The system featuring dynamic-threshold can outperform the one with the static-threshold. Consequently, the PLCAM approach is efficient and can be expected to benefit learners.

In conclusion, the three main contributions of this paper include:

(1) Proposal of the extensible CAR to represent relevant content adaptation data, which can be processed by data mining techniques.
(2) Proposal of the LCAMS to efficiently search, retrieve, and maintain the historical CARs based on the proposed CADT.
(3) Proposal of the adaptation decision process to efficiently determine a suitable content version, specifically adapted for the learner.

In the near future, we are going to deploy the PLCAM in actual mobile learning environments to offer services and to further analyze performance and learners' overall satisfaction. In addition, we will try to integrate the PLCAM with the existing content structure analysis approaches and apply it to different mobile learning scenarios to investigate its extensibility capabilities.

Acknowledgments This research was partially supported by National Science Council of Republic of China under the number of NSC 97-2511-S-009-001-MY3 and NSC 98-2511-S-009-007-MY3.

References

Basaeed, E., Berri, J., Zemerly, M.J., Benlamri, R.: Learner-centric context-aware mobile learning. IEEE Multidiscip. Eng. Educ. Mag. **2**(2), 30–33 (2007)

Beekhoven, S., Jong, U.D., Hout, H.V.: Different courses, different students, same results? an examination of differences in study progress of students in different courses. High. Educ. **46**(1), 37–59 (2003)

Buyukkokten, O., Garcia-Molina, H., Paepcke, A.: Seeing the whole in parts: text summarization for Web browsing on handheld devices. In: Proceedings of the 10th International Conference on World Wide Web, pp. 652–662, Hong Kong (2001)

Buyukkokten, O., Kaljuvee, O., Molina, H.G., Paepcke, A., Winograd, T.: Efficient web browsing on handheld devices using page and form summarization. ACM Transac. Inform. Syst. (TOIS). **20**(1), 82–115 (2002)

CC/PP.: Composite capabilities/preference profiles (CC/PP): structure and vocabularies 2.0. http://www.w3.org/Mobile/CCPP/ (2010)

Chang, H.P., Hung, J.C., Wang, C.C., Weng, M.T., Shih, T.K.: A learning content adaptation tool with templates for different handhelds. In: Proceedings of the 22nd International Conference on Advanced Information Networking and Applications, pp. 457–463 Okinawa, Japan (2008)
Chen, W., Mizoguchi, R.: Communication content ontology for learner model agent in multi-agent architecture. In: Advanced Research in Computers and Communications in Education, Proc. ICCE'99, pp. 95–102 (1999)
Chen, G.D., Liu, C.C., Ou, K.L., Liu, B.J.: Discovering decision knowledge from Web log portfolio for managing classroom processes by applying decision tree and data cube technology. J. Educ. Comput. Res. **23**(3), 305–332 (2000)
Chen, Y., Ma, W.Y., Zhang, H.J.: Detecting Web page structure for adaptive viewing on small form factor devices. In: Proceedings of the 12th International Conference on World Wide Web, pp. 225–233, Budapest, Hungary (2003)
Chen, C.S., Yang, S.J.H., Zhang, J.: Enhancing the precision of content analysis in content adaptation using entropy-based fuzzy reasoning. Expert Syst. Appl. (ESWA) **37**, 5706–5719 (2010)
Chen, L. Zhu, R., Zhou, H.: Application of fuzzy ISODATA clustering method in product identity planning. In: Proceedings of IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design, pp. 33–36, Wenzhou, China (2009)
Ding, J., Hillston, J., Laurenson, D.I.: Evaluating the response time of large scale content adaptation systems using performance evaluation process algebra. In: Proceedings of International Communications Conference (ICC2010), pp. 1–5, Cape Town, South Africa (2010)
Dewhurst, D.G., Macleod, H.A., Norris, T.A.M.: (2000) Independent student learning aided by computers: an acceptable alternative to lectures? Comput. Educ. **35**, 223–241
DOM.: Document Object Model. http://www.w3.org/DOM/ (2010)
Franzoni, A.L., Assar, S., Defude, B., Rojas, J.: Student learning styles adaptation method based on teaching strategies and electronic media. In: Proceedings of the Eighth IEEE International Conference on Advanced Learning Technologies, pp. 778–782. Santander, Cantabria (2008)
Fudzee, M.F.M., Abawajy, J.: A classification for content adaptation system. In: Proceedings of iiWAS2008, pp. 426–429, Linz, Austria (2008)
Gilbert, J.E., Han, C.Y.: Adapting instruction in search of a significant difference. J. Netwo. Comput. Appl. **22**, 149–160 (1999)
González-Castaño, F.J., Anido-Rifón, L., Costa-Montenegro E.: A new transcoding technique for PDA browsers, based on content hierarchy. In: Human Computer Interaction With Mobile Devices, pp. 69–80. LNCS, vol. 2411 (2002)
Hall, D.J., Ball, G.B.: ISODATA: a novel method of data analysis and pattern classification. Technical report, Stanford Research Institute, Menlo park, CA (1965)
He, J., Gao, T., Hao, W., Yen , I.L., Bastani, F.: A flexible content adaptation system using a rule-based approach. IEEE Trans. Knowl. Data Eng. **19**(1), 127–140 (2007)
Hinz, M., Fiala, Z., Wehner F.: Personalization-based optimization of Web interfaces for mobile devices. In: Mobile Human-Computer Interaction—MobileHCI 2004, pp. 741-748. LNCS, vol. 3160 (2004)
Hsiao, J.-L., Hung, H.-P., Chen, M.-S.: Versatile transcoding proxy for internet content adaptation. IEEE Trans. Multimed. **10**(4), 646–658 (2008)
ImageMagicK.: http://www.imagemagick.org/ (2010)
Kolb, D.A.: Learning Style Inventory: Technical Manual. McBer, Boston, MA (1976)
Kolb, D.A.: Kolb's Learning Styles. http://www.businessballs.com/kolblearningstyles.htm (2004)
Kim, W.S., Jang, D.H., Kim, T.Y.: Improved Web content adaptation for visual aspect of mobile services. In: Proceeding of the Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, pp. 402–408 (2008)
Ko, S.K., Choy, Y.C.: A structured documents retrieval method supporting attribute-based structure information. In: Proceedings of the 2002 ACM Symposium on Applied Computing, pp. 668–674. Madrid, Spain (2002)
Kobsa, A: Generic user modeling systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web, pp. 136–154. Springer-Verlag, Heidelberg (2007)
Laakko, T., Hiltunen, T.: Adapting Web content to mobile user agents. IEEE Internet Comput. **9**(2), 46–53 (2005)
Lee, S.H., Lee, J.H., Lee, E.: An inference engine for personalized content adaptation in heterogeneous mobile environment. In: Proceeding of Ubiquitous Computing Systems (UCS 2006), pp. 158–170. LNCS, vol. 4239 (2006)

Lum, W., Lau, F.: User-centric content negotiation for effective adaptation service in mobile computing. IEEE Trans. Softw. Eng. **29**(12), 1100–1111 (2003)
McIlroy, D., Bunting, B., Tierney, K., Gordon, M.: The relation of gender and background experience to self-reported computing anxieties and cognitions. Comput. Hum. Behav. **17**, 21–33 (2001)
Muntean, C.H.: Improving learner quality of experience by content adaptation based on network conditions. Comput. Hum. Behav. **24**(4), 1452–1472 (2008)
Mohan, R., Smith, J.R., Chung , S.L.: Adapting multimedia internet content for universal access. IEEE Trans. Multimed. **1**(1), 104–114 (1999)
Mohomed, I., Chin, A., Cai, J.C., Lara, E.: Community-driven adaptation: automatic content adaptation in pervasive environments. In: Proceeding of IEEE Workshop Mobile Computing Systems & Applications (WMCSA), pp. 124–133 (2004)
Mohomed, I., Cai, J.C.M., Lara, E.: URICA: UsageawaRe interactive content adaptation for mobile devices. In: Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006 (EuroSys'06), pp. 345–358, Leuven, Belgium (2006a)
Mohomed, I., Cai, J.C.M., Chavoshi, S., Lara, E.: ContextAware interactive content adaptation. In: Proceedings of the 4th International Conference on Mobile systems, Applications and Services (MobiSys'06), pp. 42–55, Uppsala, Sweden (2006b)
Mohomed, I., Scannell, A., Bila, N., Zhang J., Lara, E.: Correlation-based content adaptation for mobile Web browsing. In: Proceedings of the ACM/IFIP/USENIX 2007 International Conference on Middleware, pp. 101–120. LNCS, vol. 4834 (2007)
Mylonas, P., Tzouveli, P., Kollias, S.: E-learning and intelligent content adaptation: an integrated approach. Int. J. Contin. Eng. Educ. Life Long Learn. **17**(4–5), 273–293 (2007)
Nimmagadda, Y., Kumar, K., Lu, Y.H.: Adaptation of multimedia presentations for different display sizes in the presence of preferences and temporal constraints. IEEE Trans. Multimed. (2010) (in press)
Pettersson, O., Gil, D.: On the issue of reusability and adaptability in M-learning systems. In: Proceedings of the 6th IEEE International Conference on Wireless, Mobile, and Ubiquitous Technologies in Education, pp. 161–165, Kaohsiung, Taiwan (2010)
Queirós, R., Pinto, M.: EDUMCA: an approach to educational mobile content adaptation. In: Proceedings of 4th Iberian Conference on Systems and Information Technology (CISTI 2009), Povoa de Varzim, Portugal (2009)
Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
Rahimi, E., Shokouhi, S.B., Sadr, A.: A parallelized and pipelined datapath to implement ISODATA algorithm for rosette scan images on a reconfigurable hardware. In: Proceedings of the IEEE International Conference on Granular Computing (GRC 2007), pp. 433–433, Fremont, CA (2007)
Ramaswamy, L., Iyengar, A., Liu, L., Douglis, F.: Automatic fragment detection in dynamic web pages and its impact on caching. IEEE Trans. Knowl. Data Eng. **17**(6), 859–874 (2005)
Riding, R., Cheema, I.: Cognitive styles: an overview and integration. Educ. Psychol. **11**, 193–215 (1991)
Romero, C., Ventura, S. (eds.): Data Mining in E-Learning, Advances in Management Information. Vol. 4, WITPress, Southampton (2006)
Romero, C., Ventura, S.: Educational data mining: a review of the state of the art. IEEE Trans. Syst. Man Cybernet. C Appl. Rev. **40**(6), 601–618 (2010)
SCORM.: Sharable content object reference model 2004. In: Advanced Distributed Learning. http://www.adlnet.org/ (2010)
Smith, E.S.: The relationship between learning style and cognitive style. Personal. Individ. Diff. **30**, 609–616 (2001)
SoX.: http://sox.sourceforge.net/ (2010)
Stern, M., Woolf, P.: Adaptive content in an online lecture system. In: Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, pp. 291–300, Trento, Italy, (2000)
Su, J.M., Tseng, S.S., Wang, C.Y., Lei, Y.C., Sung, Y.U., Tsai, W.N.: A content management scheme in SCORM compliant learning object repository. J. Inf. Sci. Eng. (JISE) **21**(5), 1053–1075 (2005)
Tan, P.N., Steinbach, M., Kumar, V. (eds.): Introduction to Data Mining. Addison Wesley, New York (2005)
Tong, M.W., Yang, Z.K., Liu, Q.T., Liu, X.N.: A novel content adaptation model under E-learning environment. In: Proceedings of the 36th Annual Frontiers in Education Conference (FIE 2006), pp. 1–5, San Diego, California (2006)

Tretiakov, A., Kinshuk.: A unified approach to mobile adaptation of educational content. In: Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04), pp. 101–105, Joensuu, Finland (2004)

UAProf.: User Agent Profile (UAProf): User Agent Profiling Specification. http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf (2010)

Villard, L., Roisin, C., Layaïda, N.: An XML-based multimedia document processing model for content adaptation. In: Proceeding of 8th International Conference on Digital Documents and Electronic Publishing. LNCS, vol. 2023, pp. 104–119. Springer, Berlin, Heidelberg (2000)

Wilson, E.V.: Student characteristics and computer-mediated communication. Comput. Educ. **34**, 67–76 (2000)

Wong, E.Y.C., Chan, A.T.S., Leong, H.V.: Efficient management of XML contents over wireless environment by Xstream. In: Proceedings of the 2004 ACM Symposium on Applied Computing, pp. 1122–1127, Nicosia, Cyprus (2004)

Yang, S.J.H., Chen, Y.L., Chen, R.: Applying content adaptation technique to enhance mobile learning on blackboard learning system. In: Proceedings of the Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007), pp. 247–251, Niigata, Japan (2007a)

Yang, S.J.H., Zhang, J., Chen, R.C.S., Shao, N.W.Y.: A unit of info.-based content adaptation method for improving web content accessibility in the mobile internet. ETRI J **29**(6), 794–807 (2007b)

Yang, S.J.H., Zhang J., Huang, A.F.M., Tsai, J.J.P., Yu, P.S.: A context-driven content adaptation planner for improving mobile internet accessibility. In: Proceeding of 2008 IEEE International Conference on Web Services (ICW 2008), pp. 88–95, Beijing, China (2008)

Yin, X.Y., Lee, W.S.: Using link analysis to improve layout on mobile devices. In: Proceedings of the 13th International Conference on World Wide Web, pp. 338–344. New York, NY, USA (2004)

Zhao, X., Ninomiya, T., Anna, F., Okamoto, T.: A context-aware prototype system for adaptive learning content in ubiquitous environment. In: Proceedings of the 2008 IEEE International Symposium on IT in Medicine and Education, pp. 164–168, Xiamen (2008)

Zhu, Z., Tian, Y., Xu, J., Deng, X, Ren, X.: An improved partitioning-based web documents clustering method combining GA with ISODATA. In: Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery, pp. 208–213, Haikou (2007)

Author Biographies

Jun-Ming Su received the Ph.D. degree in Computer Science from the National Chiao Tung University in 2006 and the M.S. degree in the same field from the National Chung Hsing University in Taiwan. From 2006 to 2010, he was a Postdoctoral Researcher of the Department of Computer Science and a project leader of the E-Learning and E-Application Research Center (ELEARC) at the National Chiao Tung University in Taiwan. Currently, he is an Assistant Professor at the Department of Information and Learning Technology, National University of Tainan in Taiwan, and an Associate Executive Editor of International Journal of Digital Learning Technology (IJDLT). His current research interests include intelligent e/m/u/g-learning, expert systems, data mining, and Internet-based applications.

Shian-Shyong Tseng received the Ph.D. degree in computer engineering from the National Chiao Tung University in 1984. From 1983 to 2009, he was on the faculty of the Department of Computer and Information Science at National Chiao-Tung University, and chaired it from 1991–1992 and 1996–1998. From 1992 to 1996, he was the Director of the Computer Center at the Ministry of Education and the Chairman of Taiwan Academic Network (TANet) management committee. In Dec. 1999, he founded Taiwan Network Information Center (TWNIC) and serves currently as the Chairman of its board of directors. He joined Asia University in 2005, serving as the Dean of the College of Computer Science, and currently as a vice president. Dr. Tseng is an Editor-in-Chief of International Journal of Digital Learning Technology, an editor of International Journal of Fuzzy Systems, Journal of Internet Technology and International Journal of Computational Science. He is also a Co Editor-in-Chief of Asian Journal of Health and Information Science. His current research interests include expert systems, data mining, computer-assisted learning, and Internet-based applications. He has published more than 100 journal papers.

Huan-Yu Lin graduated with B.S. and M.S. degrees in Computer and Information Science from the National Chiao Tung University, Taiwan in 2004 and 2006, respectively. Currently, he is a Ph.D. student at Department of Computer Science, National Chiao Tung University in Taiwan. His current research interests include e-learning, knowledge engineering, expert systems, and data mining.

Chun-Han Chen received the B.S. and M.S. degree in Computer and Information Science from the National Chiao Tung University in Taiwan in 2003 and 2005, respectively. Since 2006, he has been a research and development engineer at MediaTek Inc. in Taiwan. He works in the fields of digital TV and multimedia software.