Personalized Adaptive Content System for Context-Aware Mobile Learning

Xinyou ZHAO, Fumihiko ANMA, Toshie NINOMIYA, Toshio OKAMOTO

Graduate School of Information System, the University of Electro-Communications
1-5-1, Chofugaoka, Choufu-city, Tokyo, 182-8585, Japan

Abstract

Mobile and ubiquitous computing devices are transforming the way that learners study. But most of learning contents, designed for desktop platforms, are not suitable for handheld devices. Also, some materials, irrelevant to learner's preferences or contextual environment, may affect the learning efficiency, and also increase the communication costs. In order to provide adaptive contents based on device capabilities and learner's experience, this paper presents a functional architecture for personalized adaptation contents. Also, it proposes some algorithms to create the adaptive and intelligent contents for learners. The learning contents created are adaptive to learner's preference, also adaptive to contextual environment. After the evaluation on a personalized adaptive content system developed, we find that the context-aware mobile learning system can increase the learning efficiency and interest, also resolve the new-item question during adaptation.

Key words:

Adaptive Content, Mobile Learning, Adaptive Learning, Pervasive Computing, Content Transcoding, e-learning

1. Introduction

The advances in mobile communication technologies and the rapid adoption of mobile devices with Internet capabilities make learners access the learning contents at any time, at any place, with heterogeneous mobile devices, such as hand-held PCs, personal digital assistants (PDAs), and mobile phones. These heterogeneous mobile devices are different from one to another in their screen size, memory, computing capability.

While ubiquitous devices can also complement e-Learning or traditional classroom learning, accessing e-learning contents designed for desktop platforms on a mobile device has not become as convenient as expected with mobile browser embedded in mobile device.

The problem is that most learning contents (such as standard documents, image, audio, and video) presented on a mobile device, which presentation capabilities are restrictive, may not be supported by mobile devices. Also, the mobile device may run different operation systems (Symbian, Windows Mobile, Linux) and support different markup language, such as WML, cHTML, XHTML. Hence, there is a desire to transcode the e-learning content to an adaptive format that is more suitable to be presented on the mobile devices.

There is quite a wide spectrum of implementation models for content adaptation. In regarding to the location where the adaptation is performed, W3C categorizes three types of content adaptation: server side adaptation, proxy side adaptation and client side adaptation. But due to the lack of computing power and bandwidth, the transcoding possibilities of the client side approach is very limited.

Server side approach is able to overcome the problems that the client side approach has. When a web server employs some types of device detection together with a device feature repository, it can then send optimized content to the requesting device based on its capabilities.

Proxy side adaptation is where the content is altered as it passes through one or more network components. Some network operators, for example, compress images before they are passed over the air to the mobile device.

Besides the features of mobile device, contents should also be adapted to learners' experience and preferences. The system should recommend personalized contents by exploiting other contextual

data during learning, such as learning location, time, learners' learning activity, educational strategy etc. At last, the Top-N personalized contents, which are adaptive to learner's contextual data, are recommended.

Today, many adaptive recommendation systems have been applied. But current systems only consider the features from learners separately, such as device or learners profiles. Contextual data of learners are not comprehensively computed together. Also, computing multidimensional data during adaptation will increase the computation complexity and time. So it is still a challenging working to provide adaptive contents for learners based on multidimensional contextual data.

The rest of this paper is organized as follows. In section 2, we discuss some related previous works in brief. Section 3 presents the details of the personalized adaptation system architecture for mobile learning. Section 4 and 5 give the related definitions and adaptation algorithms for adaptive system. In section 6, the performances of the personalized adaptation system are evaluated via the empirical studies on mathematics learning. Finally, this paper concludes and sketches the future research in section 7.

2. Preliminaries

2.1. Related work

There is quite a wide spectrum of implementation models for content adaptation based on user's preferences. [1] provided an algorithm to recommend Top-N contents for users based on users' history. The recommendation systems in [2] and [3] had been developed for users when users revisit the same system with similar requirements. [4] and [5] provided personalized contents by analyzing the learning log information. In [6] and [7], colony system was adopted to achieve adaptive contents for learners. Based on an ant colony optimization algorithm, style-based and attributes-based methods help learners find an adaptive learning object more effectively.

In order to provide adaptive contents based on mobile devices, [8] proposed one method to detect the mobile device capabilities and learner's preference. The personal social networks are used to detect the user's calm behavior. [9] and [10] proposed adaptation content algorithms to parser the HTML contents for mobile users. The adapted HTML contents are created at last. Also, [11] proposed a tree algorithm to negotiate the original content for transcoding. SeungMin[12] proposed one adaptation scheme for QoS-aware delivery, which can adapt multimedia contents based on current user, system and network state, to meet end-to-end requirements. Some researchers, [2][13], had also adopted caching strategy, to reduce the transcoding time. When caching content stratifies with transcoding requirement, the server returns the caching contents directly. But this method also increases the storing burden of server.

Learner's context data are also considered by [14][15][16]. These methodologies can combine the learning infrastructure (educational activity, networks, learning state, etc) to identify the interest & preference of learners and create adaptive contents for learners in a ubiquitous environment.

2.2. Device Profile

Ubiquitous mobile devices usually differ in hardware, software, and browsing capabilities. The ability of transcoding the content according to the device capability and learner's preference and experience is regarded as context awareness [17], which includes three important aspects of context: (1) where you are; (2) who you are with; and (3) what resources are nearby.

Context awareness of mobile learner can be achieved for each learner based on device profile describing the capabilities after the adaptation system receives http request from mobile device. Composite capabilities/preferences profile (CC/PP) [18] and User Agent Profile (UAProfile) [19], developed by W3C and OMA separately, are the prevalent standards of describing the device capability. The specifications of CC/PP and UAProfile follow the standard of resource description framework (RDF), which is based on XML and has a tree structure to record the value of each feature [20].

There are millions of mobile devices existing today. In general, these device profiles are stored on the server of device manufacture. So we have to collect each profile from these device manufactures for our applications. Fortunately, Luca has defined a Wireless Universal Resource File (WURFL) [21] model to describe the features of mobile devices and browsers. In terms of adoption, WURFL is today more popular than pure UAProf or CC/PP solutions. WURFL model is an XML configuration file which contains information about capabilities and features of many mobile devices in the wireless world. Also, the repository of device in WURFL is updated everyday by contributors in the world.

3. Adaptation Contents Architecture

In our research, we divide the system into four engines: Leaner Context Engine (LCE), Detector Engine (DE), Adaptive Content Delivery Engine (ACDE), and Transcoding Engine (TE). The architecture is shown in Figure 3-1. After ACDE receives the context data and features of device, it creates adaptive contents for mobile learners.

Detector Engine takes responsibility for detecting all capabilities (memory, screen size …) using the WURFL (Wireless Universal Resource File) model to define the features of devices and mobile browsers. Please refer to [8] for details about detecting algorithms. Learner Context Engine detects the contextual information of learners, such as location, time, network performance, schedule, weather etc.

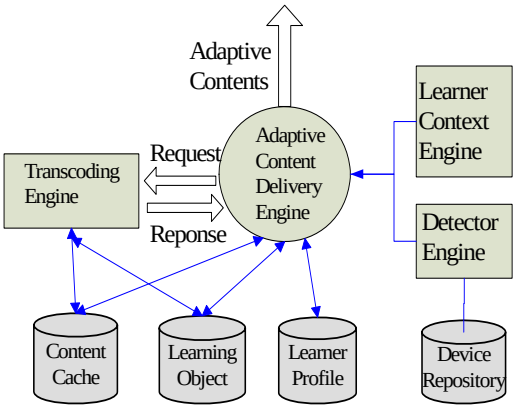


Figure 3-1 Adaptation Content Architecture

Transcoding Engine (TE) contains many different conversion engines for media, such as text, image, standard document, audio, and video. After TE parses transcoding request (XML) from adaptive content delivery engine (ACDE), it choose a conversion engine to transform contents into adaptive content and responses the transcoding result (XML) to ACDE. In our research, we adopt the OMA standard XML [22] format for transcoding job request and job result.

ACDE is a most important component in adaptive system. It recommends the adaptive Top-N contents based on learner's contextual data: content features, device capabilities, and learner's preference and experience. The recommended contents are adapted to learner, also to mobile device. A big size video, for example, will not be recommended because it is impossible to display on a mobile phone.

When one of the recommended contents is accessed by a learner, ACDE judges whether it can be displayed on user's device correctly after ACDE analyzes the features of all parts from content. If not, ACDE negotiates the content and sends a request job (contains some parts of content not displayed) to transcoding engine for adaptation. The Figure 3-2 shows one example for transcoding job request and job result.

Finally, ACDE sends the transformed parts with other parts of content embedded with the adaptive markup language to mobile learner. To mobile markup language in this research, the cHTML, WML and XHTML MP are mainly considered.

```
- <transcodingJob>
    <jobID>Audio_1902</jobID>
  - <source>
      <contentType>audio/x-mpeg</contentType>
      <location>source/au0801120103.mp3</location>
    </source>
  - <target>
    - <transcodingParams>
        <contentType>audio/amr</contentType>
        <samplingRate>8000</samplingRate>
        <channels>1</channels>
        <bitRate>12200</bitRate>
      </transcodingParams>
    </target>
  </transcodingJob>
- <jobResult>
    <jobID>Audio_1902</jobID>
  - <mainReturnResult>
      <returnCode>2000</returnCode>
      <returnString>Success</returnString>
    </mainReturnResult>
    <duration>30</duration>
  - <output>
      <contentType>audio/amr</contentType>
      <location>tmp/au0801120103_1202.amr</location>
    </output>
  </jobResult>
```

Figure 3-2 Transcoding Request & Result

4. Related Definitions

4.1. Device Capabilities

Device capabilities contain most features of device, which is described in Formula (4-1) for device D_i:

$$D_i = (f_{i1}, f_{i2}, f_{i3}, ..., f_{in}); \quad (4\text{-}1)$$

Where, f_{ij} is one of all features of mobile device i; n is total count of capabilities of device considered in system.

4.2. Contextual Data

In our research, we mainly consider 6 contextual elements in learning environment: the location, time, network performance, network type, and mobile device type. Contextual data (E_i) for learner i is expressed in Formula (4-2).

$$E_i = (l_i, t_i, u_i, d_i, p_i, m_i); \quad (4\text{-}2)$$

Where, l_i: location (campus, car, train, road, classroom, home); t_i: time; u_i: upload speed by mobile device; d_i: download speed by mobile device; p_i: network type (GSM, 3G, IEEE802); m_i: mobile device type (PC, PDA, Smartphone, Mobile Phone).

For example, a learner access learning content by his GSM phone in the moving car at 10:30. The E = (car, 10:30, 6k, 24k, GSM, Phone).

4.3. Content's Item

Each learning content (C) contains many different items, such as a web page may contains text information, image, and audio, which is described in Formula (4-3).

$$C = (I_1, I_2 ,\ldots, I_n) \quad (4\text{-}3)$$

Where, I_i is one item of the learning content (C); n is total different item in the learning content.

Each item (I_i) of learning content has two vectors: feature vector of item and contextual feature vector of item, which is described in Formula (4-4).

$$I_i = (fv_i, cfv_i), \quad (4\text{-}4)$$

Where, $fv_i=(f_{i1}, f_{i2}, f_{i3}, \ldots, f_{in})$, each f_{ij} in vector represents a feature value of item i; $cfv_i =(w_{i2}, w_{i2}, \ldots, w_{im})$, each w_{ij} in vector represents a weight for each contextual data; n is the total count of item features; m is the total count of contextual data considered in system.

4.4. Learner's Preference

Each learner's preference (P_i) is described in a vector in Formula (4-5):

$$P_i = (w_{i1}, w_{i2}, \ldots, w_{in}) \quad (4\text{-}5)$$

Where, each w_{ij} in vector represents a weight for each contextual data; n is total count of contextual data related to learners.

4.5. Similarity Algorithm

In our research, we compute the similarity between two learners or two items in Dice Coefficient Similarity in Formula (4-6).

$$Sim(U_t, U_q) = \frac{2P(U_t, U_q)}{P(U_t) + P(U_q)} \cong \frac{\sum_{k=1}^{n} w_{kt} w_{kq}}{\sum_{k=1}^{n} w_{kt}^2 + \sum_{k=1}^{n} w_{kq}^2} \quad (4\text{-}6)$$

Where, U_t and U_q are weights vector for learner or item at a learning state; w_{ki} is the weight of one attribute; n is length of vector.

5. Adaptation Algorithms

The adaptation and intelligent process contains three steps: adaptive contents recommendation, content negotiation, and content transforming. The algorithm is shown in Table 5-1.

Table 5-1 Adaption Algorithm

1. Find the Top-M Neighbor Set by similarity algorithm between active user and other users' preference and contextual data and histories;
TNS = TN(P, D, E)
2. Find the Top-K contents set from Top-M's neighbors for active user at contextual data;
TCS = TK(P, C_i, E)
3. Compute the similarity among Top-K and all contents and recommend Top-N contents for active user;
$s_{i,j}$ = Sim (C_i, C_j)
S = {$s_{i,j}$}, i = K, j =total number of Contents
R = {r_i| r_i >= each s $\in$ S – R, $r_i \in$ S)
4. Negotiate the content accessed by learners and recommend the appreciate version for mobile user;
V = N(D, E, C$\in${R})
5. Transform contents into adaptive contents for user.

$$A = T(V, C)$$

5.1. Adaptive Contents Recommendation

Adaptive contents recommendation contains three steps: find Top-M Neighbors Set (TNS) with active learner, recommend Top-K contents set (TCS) among TNS, and recommend Top-N contents set for active learner.

In algorithm TN and TK, the system should consider multidimensional data: device, learner preference, item, and contextual data, which increase the computing complexity. In our research, we only consider the mobile device type (at this moment, other features, such as image size, do not affect the adaptive process) during TN. So we combine E_i and P_i into $L_i = (E_i, P_i)$. Then we may use Formula (4-6) to cluster the learners. The same method is applied for TK.

As we know, it is difficult to recommend new item because they have a low reading rate [3]. In order to resolve the new item question, the system re-computes the similarity among Top-K and other items. Last, the system may recommend Top-N contents R based on all items stored in server.

5.2. Content Negotiation

The negotiation algorithm is shown in Table 5-2. The system extracts the different parts from original content accessed by learner, such as TEXT, IMAGE, and AUDIO. Also, features of each item are indentified, $I_i = (fv_i, cfv_i)$.

Table 5-2 Content Negotiation Algorithm

1. Parse the contents into different parts, text, image, audio,….

$$C = \{I_i | I_1 \cup I_2 \cup \ldots \cup I_n = \text{Contents}\}$$

2. Get the features of each I_i,

$$I_i = (fv_i, cfv_i)$$

3. Create a Feature Finite State Machine ($FFSM_i$) for I_i, the initial state contains features about I_i, branches are constrained attribute and other state represents the constrained features $FFSM_i$ (stored in a XML file);

4. Stop if the $FFSM_i$ satisfies the constrained condition

To I_i, we use feature finite state machine ($FFSM_i$) to negotiate the final constrained limitation. The model of FFSM is shown in Figure 5-1. F is initial state and F' is final state. Each directed edge represents a limited capability condition of mobile device, such as size limitation of image from state0 to state2. Another, contextual feature vector (cfv_i) in I_i is also considered by Formula (4-6). At last, the system may create a XML file for transcoding engine (an example for transcoding job shown in Figure 3-2).

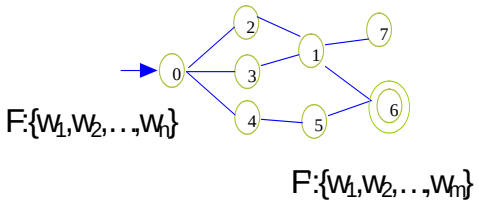


Figure 5-1 FFSM Model for Negotiation

5.3. Transcoding Algorithm

The system adopts the FFMPEG[23] and ImageMagick[24] to transform the video & audio and image separately based on constrained features from request. To Standard Documents (PPT, DOC, PDF etc), we have developed an agent to extract contents (IMAGE+TEXT) for mobile learners.

The transcoding algorithm is shown in Table 5-3. For example, the transcoding command for example in Figure 3-2: *ffmpeg -i source/au080 1120103.mp3 -ac 1 -ab 11200 -ar 8000 -f amr au0801120103_ 1202.amr*.

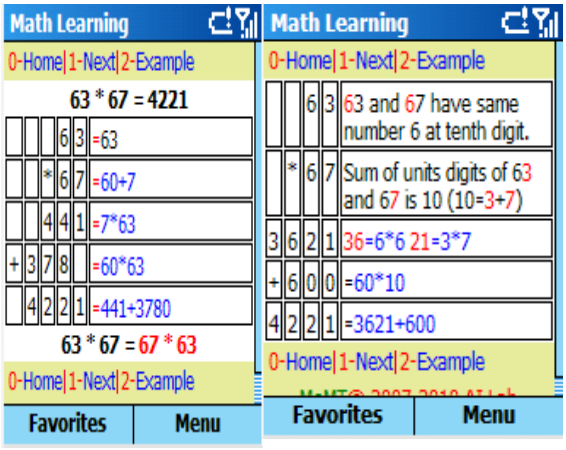
Table 5-3 Transcoding Method

```
get the constrained V features from XML
if ( V of Item existed in pre-cached database )
  pointer = XML file of pre-cached content;
else
     switch(type of media)
       case image:    transform based on V;
       case text:     transform based on V;
       case audio:    transform based on V;
       case video:    transform based on V;
       case document: transform based on V;
    end switch
  pointer = XML file of transformed content;
end if
```

6. Architecture Evaluation

A Mobile Mathematics Tutoring (MoMT) system has

been developed for primary students based on the system architecture above. It supports mathematics learning with mobile devices at any time and any place. MoMT analyzes learner context and provides customized tutoring based on the learner's knowledge. In Figure 6-1, Figure A is an example for general tutoring and Figure B is an advanced tutoring for high level student.



A B

Figure 6-1 Customized Tutoring

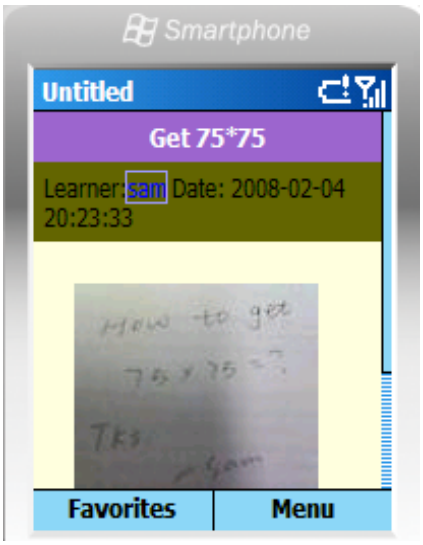


Figure 6-2 Adaptive Image on Device

Learners may send questions (including text message, photos and video/audio captured by cameras or recorders embedded in mobile device) by mobile email or mms to participant the discussion in mobile device. Thus, MoMT combines text messages with photos and video/audio to foster rich contexts for discussion. The Figure 6-2 shows one image taken by Sharp 705H phone (480x640) on a smart phone.

6.1. Device Context Awareness

We have tested among mobile phone, PDA, smart phone, iPod, and simulator. Of course, we also test some devices not included in the device repository. The result is shown in Table 6-1. The system can detect the attribute values based on the need of MoMT from PDA, smart phone, iPod, and simulator. But we may not detect the features from mobile phone at some times. At this situation, we replace the capabilities by a similar device. The Figure 6-3 shows the detecting results based on algorithms.

Table 6-1 Device Context

Type	Total Count	Screen size	Browser	Image format
Mobile Phone	68	66	65	61
PDA	3	3	3	3
Smart Phone	1	1	1	1
iPod	1	1	1	1
Simulator	2	2	2	2

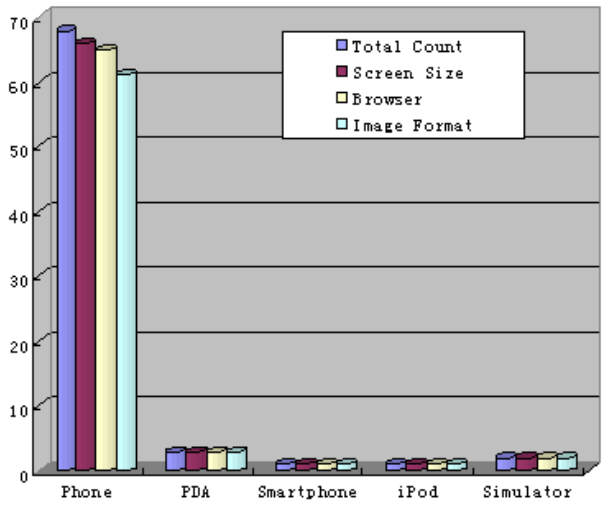


Figure 6-3 Device Context Awareness

6.2. Transcoding Performance Measurement

Transcoding engine is responsible for transforming

the learning contents into suitable format for mobile device. In this paper, we present the image transcoding based on context awareness in MoMT.

The Table 6-2 shows the data size of original and transformed image with different screen size and network speed. The system can transcode the image into adaptive image for mobile learner. The Figure 6-4 shows the results of image transcoding.

Table 6-2 Image Transcoding

Device	Screen Size	Colors	Size	Speed	Ratio
O*	600x390	16777216	18,580	100M	100%
D_1**	240x320	65536	8,046	54M	43.30%
D_2**	240x320	65536	8,046	11M	43.30%
S_1***	240x320	262144	4,877	10k	26.25%
S_2***	240x320	262144	1,236	1k	6.65%

* O: Original Image on PC
** D: Dell PDA x50v
*** S: Sharp 507SH

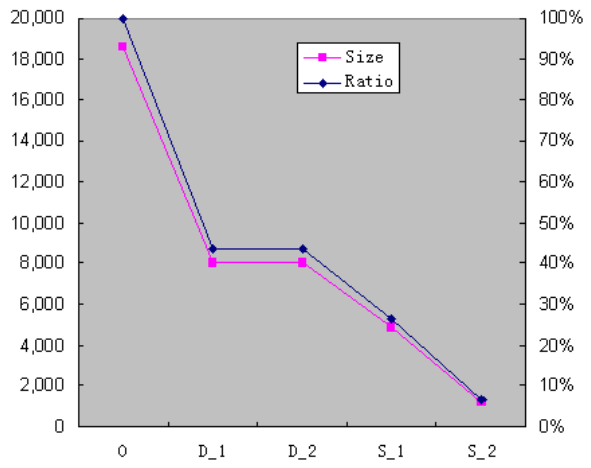


Figure 6-4 Ratio & Data Size of Image

6.3. Learning Efficiency

In order to evaluate the learning efficiency of adaptive contents to learners, we choose 10 primary learners (similar level) to attend the experiment. We divide them into two groups: Mobile (M) and Textbook (T). The group M, including 5 learners, reviews the mathematics by textbook and MoMT, and the other group T only uses the textbook for review after class.

The test contents are the two-digit multiplications and are completed in two days. We request the members in group M use the MoMT over 30 minutes. After the test, we summarize the result shown in Table 6-3.

Table 6-3 Learning Result

Learner		T1	T2	T3	Score
Group M review by Mobile and Textbook	A	32	30	17	92
	B	36	34	20	87
	C	31	32	17	85
	D	35	38	19	96
	E	37	34	20	88
Group T review by Textbook	A	0	60	20	86
	B	0	45	20	80
	C	0	43	20	79
	D	0	64	16	94
	E	0	67	18	90

* T1: Learning Time in MoMT
*T2: Learning Time in Textbook
*T3: Test Time

The mean score, score variance, and mean time of total learning and test are shown in Figure 6-5. From the Figure 6-5, we can know that learners in group M may get a better score than those in group T. We also know that the system may improve the learning interest (more 10 minutes) by the personalized learning system.

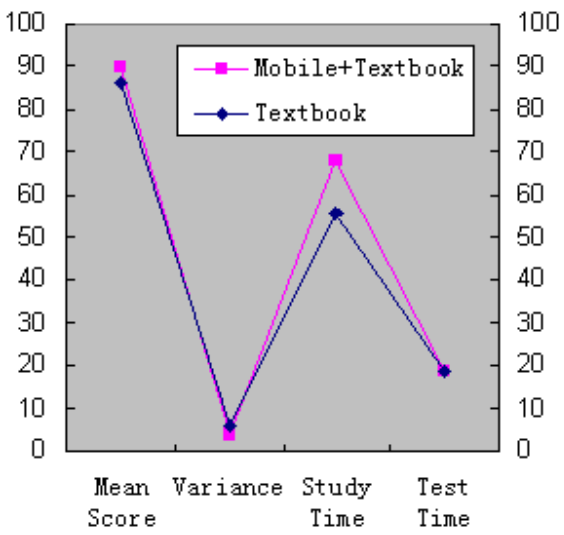


Figure 6-5 Learning Efficiency

Another, we have checked the answers from test papers. We found that some quicker, easier ways were adopted by learners in group M. This is why they complete the test earlier. Also, the attendees did a questionnaire about the MoMT. The results show that the learners feel convenient and easy to use when they need help.

7. Conclusions & Future Works

In order to resolve the diversity of learning contents and mobile device, this paper proposed adaptive and personalized system architecture for ubiquitous learning environment. Also, new items are also recommended based on clustering technology and Top-K technologies. Last, adaptive contents are created for learners based on learners' experience and device capabilities.

In future, we plan to run the system on real e-learning system to evaluate the features in ubiquitous learning environment. Also, some more contextual information will be tested on system to make it more consistent with learners' experience.

References

[1]. Mukund Deshpande and George Karypis. Item-Based Top-N Recommendation Algorithms. ACM Transactions on Information Systems. 2004, 22(1):143-177.

[2]. G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin. Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. ACM Transactions on Information Systems. 2005, 23(1):103-145.

[3]. Claudia Niederée, Avaré Stewart, Bhaskar Mehta and Matthias Hemmje. A Multi-Dimensional, Unified User Model for Cross-System Personalization. The Proceeding of the AVI 2004 Workshop on Environments for Personalized Information Access. 2004:34-54.

[4]. Toshio Okamoto and Toshie Ninomiya. Organisational knowledge management system for e-learning practice in universities. The sixth conference on IASTED International Conference Web-Based Education. 2007(10):528 - 536.

[5]. Toshie Ninomiya, Haruhiko Taira, Toshio Okamoto. A personalised learning environment architecture for e-Learning. The sixth conference on IASTED International Conference Web-Based Education. 2007(10):517 - 521.

[6]. Tzone-I. Wang, Kun-Te Wang and Yueh-Min Huang. Using a style-based ant colony system for adaptive learning. Expert Systems with Applications. 2008 (5), 34(4):2449-2464.

[7]. Yao Jung Yangaand Chuni Wua. An attribute-based ant colony system for adaptive learning object recommendation.Expert Systems with Applications (2008),doi:10.1016/j.eswa. 2008.01.066.

[8]. Xinyou Zhao, Toshio Okamoto. A Device Independent System Architecture for Adaptive Mobile Learning. The 8th IEEE International Conference on Advanced Learning Technologies (ICALT'08). 2008(7):23-25.

[9]. Ariel Pashtan, Shriram Kollipara and Michael Pearce. Adapting Content for Wireless Web Services. Internet Computing. 2003(9), 7(5):79-85.

[10]. Timo Laakko and Tapio Hiltunen. Adapting Web Content to Mobile User Agents. Internet Computing. 2005(3), 9(2):46-53.

[11]. Wai Yip Lum and Francis C.M. Lau. On Balancing Between Transcoding Overhead and Spatial Consumption in Content Adaptation. The 8th annual international conference on Mobile computing and networking.2002: 239–250.

[12]. SeungMin Rho, JeongWon Cho and EenJun Hwang. Adaptive Multimedia Content Delivery in Ubiquitous Environments. International Workshops on Web Information Systems Engineering (WISE2005). 2005(11):43-52.

[13]. Chunhong Li, Guofu Feng, Wenzhong Li, Tiecheng Gu Sanglu Lu and Daoxu Chen. A Resource-Adaptive Transcoding Proxy Caching Strategy. Frontiers of WWW Research and Development.2006:556-567.

[14]. Economides, A.A. Adaptive Mobile Learning. The Fourth IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education. 2006(11): 26-28.

[15]. N. Uday Bhaskar, P. Govindarajulu. A Design Methodology for Acceptability Analyzer in Context Aware Adaptive Mobile Learning Systems Development. International Journal of Computer Science and Network Security. 2008, 8(3):130-138.

[16]. Bryan Patten, Inmaculada Arnedillo Sánchez and Brendan Tangney. Designing collaborative, constructionist and contextual applications for handheld devices. Computers & Education.2006 (4), 46(3):294 - 308.

[17]. Jung-Lee Hsiao, Hao-Ping Hung, and Ming-Syan Chen. Versatile Transcoding Proxy for Internet Content Adaptation. IEEE Trans. on Multimedia. 2008(6). Vol.10 (4):646-658.

[18]. W3C. Composite capabilities/preferences profile, viewed on July 2007.

[19]. OMA. User Agent Profile: Version 2.0, viewed on February 2007.

[20]. W3C. Resource Description Framework (RDF) Model

and Syntax Specification, viewed on July 2007.
[21]. WURFL. http://wurfl.sourceforge.net/, viewed on September 2007.
[22]. OMA. Standard Transcoding Interface Specification (2007). http://cms.openmobilealliance.org/technical/release_program/sti_archive.aspx.2007.
[23]. FFMPEG. http://ffmpeg.mplayerhq.hu/ viewed on February 2008.
[24]. ImageMagick. http://www.imagemagick.org viewed on December 2007.

Xinyou ZHAO received his B.Sc. in Computer Science Education Technology from Xinyang Normal University, china in 2000 and Master Degree in Department of Computer Science from Guilin University of Electronic Technology, China in 2003. He worked as an instructor in Guilin University of Electronic Technology from 2003. From 2005 to 2006, he was a researcher in Waseda University.

He is currently a PhD candidate at The University of Electro-Communications, Japan. His research interests include mobile learning, data mining, intelligent tutoring system and multimedia technology. He is a member of IEEE.

Fumihiko ANMA received his B.E. from Tokyo Institute of Technology, Japan in 2000 and his Master Degree and PhD Degree from Shizuoka University, Japan in 2002 and 2005 respectively.

He is currently an assistant professor with the Graduate school of Information Systems, The University of Electro-Communications, Japan. His research interests include Knowledge Computing, E-Learning, Artificial Intelligence, and Software Agent. He is a member of Japanese Society for Information and Systems in Education, Japanese Society for Artificial Intelligence.

Toshie NINOMIYA graduated from Department of Science and Technology, Keio University in 1993 and then worked in Toyota Motor Corporation as an engineer of industrial technology.

She is currently an assistant professor in Artificial Intelligence and Knowledge Computing Lab, The University of Electro-Communications, Japan. Now her research interests include Personalized Adaptability of e-Learning Environment and Intelligent Recommendation System. She is a member of the JSISE, and the IEICE.

Toshio OKAMOTO received his B.Sc. and Master of Psychology from Kyoto University of Education, Japan in 1971 and 1973 respectively and his PhD Degree from Tokyo Gakugei University, Japan in 1975. From 1981, he worked in Tokyo Gakugei University as an instructor and became an associate professor in 1983.

He is currently a Professor and the Chairman of Center for Developing E-Learning in The University of Electro-Communications. His research interests include Intelligent Knowledge, Educational Strategy, E-Learning, e-pedagogy, and collaborative learning. He is a member of the AAAI, the IASTED, the JSISE, the IEICE, the JSET, the IPSJ and the JAEP.