Fuzzy Recommendation towards QoS-Aware Pervasive Learning

Zhiwen Yu[1,2], Norman Lin[1], Yuichi Nakamura[2], Shoji Kajita[1], and Kenji Mase[1]

[1] *Information Technology Center, Nagoya University, Japan*
zhiwen@itc.nagoya-u.ac.jp, nlin@arch.itc.nagoya-u.ac.jp, {kajita, mase}@nagoya-u.jp
[2] *Academic Center for Computing and Media Studies, Kyoto University, Japan*
yuichi@media.kyoto-u.ac.jp

Abstract

Pervasive learning promises an exciting learning environment such that users can access content and study them at anytime, anywhere, through any devices. Besides delivering the right content to the learner, it is necessary to provide acceptable Quality-of-Service (QoS) guarantees in terms of presenting the content. In this paper, we propose a recommendation approach based on fuzzy logic theory towards QoS-aware pervasive learning. It determines appropriate presentation form of the learning content according to user's QoS requirements and device/network capability. We also propose an adaptive QoS mapping strategy, which dynamically sets quality parameters at running time according to the capabilities of client devices. The experimental results show the proposed approach is feasible and acceptable to enable QoS-aware pervasive learning.

1. Introduction

The mergence of pervasive computing and e-learning makes learning at anytime, anywhere possible. For better learning experience, we need to recommend the right learning content in the right form to the right person according to the user's changing context. The user's learning context can be divided into two categories: personal context and technical context. The personal context describes context of the user's own, e.g. prior knowledge, goals, learning style, schedule, etc. The technical context, on the other hand, depicts the physical running infrastructure, e.g. terminal capability, network condition, etc. Several systems have been proposed for selecting the right learning content according to user's personal context, e.g. LIP [1] and Elena [2]. The presentation form of the learning content depends not only on the technical context, but on user's QoS (Quality-of-Service) requirements. For example:

Harry (a student) and Victoria (a teacher) are currently accessing the same content through a low-speed mobile network. Harry wants to get the content immediately as his learning task is urgent, while Victoria wishes to obtain a high quality material for preparing her lecture in the next day. So Harry may get the low quality content quickly, and Victoria may get the high quality content even if she has to wait for a long time for downloading.

The existing systems would usually present Harry and Victoria with content in the same format, i.e. regular content without any adaptation or simply send them low quality variations because of the low-speed mobile network. However, this can not satisfy both of the users.

Previous systems usually address the problem of presenting content on different pervasive devices from the technical point of view. They determine the presentation form merely based on the device capabilities, e.g. displaying capability and memory [3], sometimes combined with network features [4, 5]. They seldom consider user's requirements of Quality-of-Service (QoS) from the perspective of users themselves. In previous work, the low-quality image is by default displayed when the network bandwidth is very low. But, usually this is not what the user expects in the field of pervasive e-learning. For example, in video courseware streaming, if the content contains important text, e.g. lecturer's writing on blackboard, the learner might desire high resolution image even if the bandwidth is very low and he has to wait for a few minutes.

The dynamic pervasive environmental context and user's QoS requirements pose challenges to select the right presentation for the target user. In this paper, we

21st International Conference on Advanced Networking and Applications(AINA'07)
0-7695-2846-5/07 $20.00 © 2007 IEEE

propose a recommendation approach based on fuzzy logic theory [6] towards QoS-aware pervasive learning. It determines appropriate presentation form of learning content according to user's QoS requirements (e.g. time (urgent or not) and quality of content) and device/network capability (e.g. format, frame size, and available bandwidth). We also propose an adaptive QoS mapping strategy, which dynamically sets quality parameters at run-time according to the capabilities of client devices.

The paper is structured as follows. Section 2 describes related work. In Section 3, we present the design of the overall system. Section 4 describes the fuzzy recommendation to generate appropriate content QoS level. In Section 5, we present the dynamic QoS mapping strategy. The implementation details and experimental results are given in Section 6. Finally, Section 7 concludes the paper and points out directions for future work.

2. Related Work

Several systems, e.g. [4] and [7], adopt a rule-based approach to induce content presentation form. However, previous rule based approaches usually use crisp rules. We believe that crisp rules are too rigid to make self adaptation. It cannot intelligently simulate human's flexible inference. For example, it is difficult to distinguish: how fast is a high speed network? How much time is time-critical? We may say if the network bandwidth is above 512kbps, it is nearly high speed, 80% belongs to high speed. The description of high speed network and low level delay is quite vague. To deal with the ambiguity in pervasive learning and enable the recommendation to be made more flexible, our approach adopts fuzzy rule to interpret context and QoS requirements.

Cao et al. [8] propose a generic fuzzy-based service adaptation model (FSAM) in context-aware mobile computing middleware. It selects service policies based on the distance of fuzzy status between the policy and the current context situation. It does not provide QoS guarantees from the perspective of users themselves.

Minoh et al. [9] determine the resolution and frame rate of distance learning courses based on content, e.g. important text or general image is included in the content. The purpose of determining presentation form of learning content is similar to us; however we address the problem from the user's perspective and take into account the device capability and network condition.

QCompiler [10] provides a programming framework for quality-aware ubiquitous multimedia applications, through which a user can request different QoS level. The authors in [11] also use fuzzy control theory for QoS adaptation in distributed multimedia applications. However both of them do not consider the device capability and network bandwidth when setting the QoS level and QoS mapping is static, hence sometimes the QoS could not be guaranteed.

3. System Design

We first give the definition of the QoS concepts used in our approach.

Definition 1 (Content QoS): Content QoS represents the set of characteristics of a learning content necessary to fulfill user satisfaction. In this paper, we mainly focus on application-level parameters, e.g. frame size, frame rate, etc.

Definition 2 (Service QoS): Service QoS represents the set of characteristics of a pervasive e-learning system necessary to present the content to the user and fulfill user satisfaction. The service QoS describes not only the characteristics of the content, but the features of the application, e.g. performance-oriented (e.g. response time) and cost-oriented parameters (e.g. copyright fee).

For the service QoS, we mainly consider the response time and quality of content. The response time is also called user's waiting time. For video, audio, and flash, it is the initial buffering time between the user's request and the beginning of content display on the client. On the other hand, if the content is image, document, and text, the response time refers to the downloading time. There naturally exists conflict between the response time and quality of content. For instance, wanting the system to present the content quickly may decrease the quality of it. So the user has to balance between the two QoS parameters, i.e. which one is more important for his situation.

We therefore can use response time to represent user QoS requirements, which also implies the requirements for quality of content. If the user prefers to obtain high quality content, he can set the response time longer, which means using time to compensate for quality. Increasing the buffering time is useful to improve the quality of media streaming, especially when the network condition is not very good. On the contrary, if the user wants to get the content quickly, which implies he does not care about the quality, he can set the response time shorter.

Figure 1 illustrates the schematic architecture of the system. It basically consists of two steps: Fuzzy Recommendation and QoS Mapping.

The Fuzzy Recommendation takes network bandwidth and user's QoS requirements (response time) as inputs. The response time here implies the requirements of quality of content. As a result, it generates appropriate content QoS level through a fuzzy control process. The Fuzzy Recommendation is

0-7695-2846-5/07 $20.00 © 2007 IEEE

composed of four collaborating components. The Knowledge Base defines membership functions and fuzzy rules. The Fuzzification transforms crisp inputs into membership values. The Inference Engine performs reasoning based on the fuzzy rules. The Defuzzification transforms the fuzzy result of the inference into a crisp output.

The QoS Mapping dynamically maps the content QoS level to machine understandable QoS categories according to the capabilities of client device. It finally decides the presentation form of the learning content.

The Fuzzy Recommendation and QoS Mapping are described in greater details in Section 4 and 5.

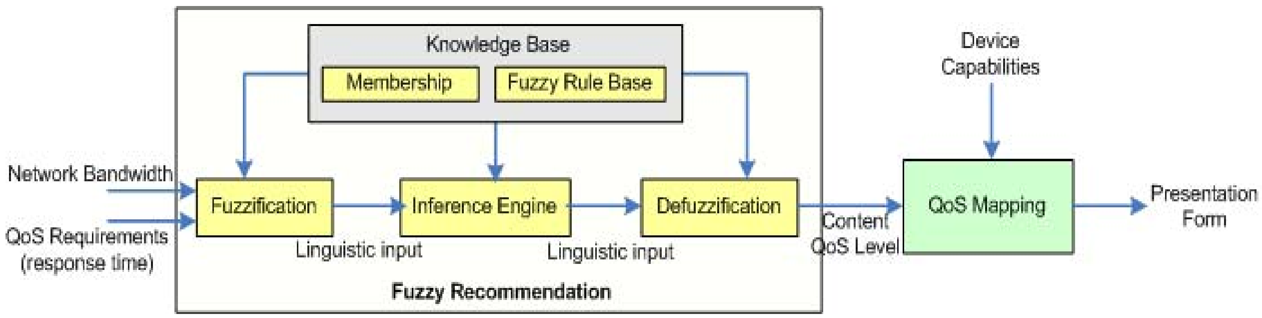


Figure 1. System architecture

4. Fuzzy Recommendation

4.1. Recommendation Process

The fuzzy recommendation is based on fuzzy logic control theory. It consists of the following four steps:

(1) Defining the membership functions for the input and output.

In the recommendation process, we set network bandwidth and desired response time as input, with content QoS level as output. The fuzzy membership functions of the network bandwidth and response time are described in Figure 2. In order to measure the bandwidth and response time universally, we normalize them into the same standard scale of [0, 1], according to the following equations.

$$E_1(network\_bandwidth) = \begin{cases} 1 & \text{if } network\_bandwidth \geq 512kbps \\ \dfrac{network\_bandwidth}{512kbps} & \text{if } network\_bandwidth < 512kbps \end{cases} \quad (1)$$

$$E_2(response\_time) = \begin{cases} 1 & \text{if } response\_time \geq 60s \\ \dfrac{response\_time}{60s} & \text{if } response\_time < 60s \end{cases} \quad (2)$$

The bandwidth and response time are classified into three sets, respectively. Each particular value may belong to 1 or 2 sets with corresponding membership degree. For example, $\mu_{Network\_bandwidth=high}(0.8)=0.8$, $\mu_{Network\_bandwidth=medium}(0.8)=0.2$ means the network bandwidth, 0.8 (i.e. 409.6kbps) belongs to high speed with confidence value of 0.8, while 20% belongs to medium speed.

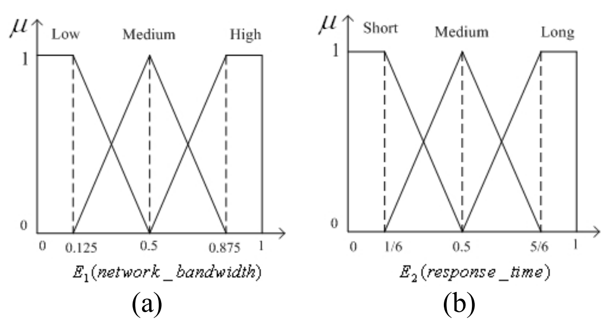


Figure 2. The fuzzy membership functions of: (a) network bandwidth, and (b) response time

The fuzzy membership function of the output, i.e. content QoS level is defined in Figure 3. It is represented with five levels or five sets with respect to fuzzy theory, namely Very low, Low, Medium, High, and Very high.

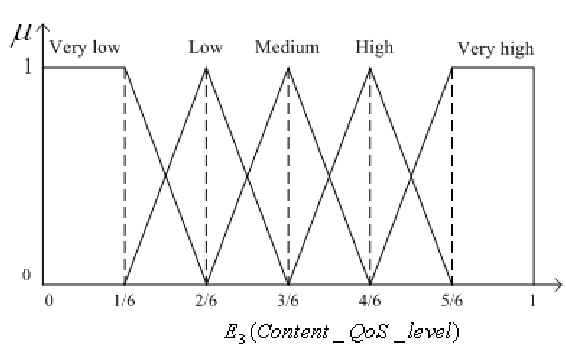


Figure 3. The fuzzy membership function of output

variable (Content QoS level)

(2) Fuzzification: mapping a particular network bandwidth and response time to the fuzzy membership correspondingly.

By using the membership functions defined above, we translate the input crisp values of network bandwidth and response time into a set of linguistic values and assign a membership degree for each linguistic value.

(3) Getting the linguistic values of content QoS level.

The inference engine performs decision-making based on the fuzzy logic inference rules. Each rule is an IF-THEN clause in nature, which determines the linguistic value of content QoS level (E_3) according to the linguistic values of network bandwidth and response time (E_1 and E_2). Based on analysis, the fuzzy inference rules are set as shown in Figure 4. For example, the first rule infers the content QoS level as Very low in the condition that the available bandwidth is Low and user's desired response time is Short. But the content QoS level will be higher, if the user is willing to wait for a longer time, as indicated by rule 2 and 3.

1. If E_1 is "Low" and E_2 is "Short", then E_3 is "Very low";
2. If E_1 is "Low" and E_2 is "Medium", then E_3 is "Low";
3. If E_1 is "Low" and E_2 is "Long", then E_3 is "Medium";
4. If E_1 is "Medium" and E_2 is "Short", then E_3 is "Low";
5. If E_1 is "Medium" and E_2 is "Medium", then E_3 is "Medium";
6. If E_1 is "Medium" and E_2 is "Long", then E_3 is "High";
7. If E_1 is "High" and E_2 is "Short", then E_3 is "Medium";
8. If E_1 is "High" and E_2 is "Medium", then E_3 is "High";
9. If E_1 is "High" and E_2 is "Long", then E_3 is "Very high".

Figure 4. Fuzzy rules

(4) Defuzzification: transferring the linguistic values of content QoS level into a crisp value and deciding the final QoS level.

The most common defuzzification methods are "center of gravity" and "mean of maximal value". The method of "center of gravity" takes more useful factors into consideration, so we adopt this method to get crisp value of content QoS level. The method of "center of gravity" is presented as follows:

$$Content\_QoS\_level = \frac{\sum_{i=1}^{n}(\mu[i] \times y_i)}{\sum_{i=1}^{n}\mu[i]} \quad (3)$$

where:

- $\mu[i]$ is the height of output area from the *i*-th rule,
- y_i is the gravity's horizontal coordinate of output area from the *i*-th rule,
- *n* is the total number of matching rules for given values of E_1 and E_2.

With the crisp value of content QoS level, we map it into its fuzzy membership and choose the linguistic value whose membership degree is the largest as the final QoS level.

4.2. Example

In this section, we give an example to demonstrate the fuzzy recommendation process.

Suppose the available bandwidth is 400kbps, and the user specified response time is 18s.

Step 1: Normalize the input
E_1(network_bandwidth)=400kbps/512kbps=25/32
E_2(response_time)=18s/60s=0.3

Step 2: Map the normalized network bandwidth and response time to the fuzzy membership defined in Figure 2

The network bandwidth, 25/32 is between "Medium" and "High", $\mu_{Network\_bandwidth=medium}(25/32) = 0.25$, $\mu_{Network\_bandwidth=high}(25/32) = 0.75$. The response time, 0.3 is between "Short" and "Medium", $\mu_{Response\_time=short}(0.3) = 0.6$, $\mu_{Response\_time=medium}(0.3) = 0.4$.

Step 3: Get the linguistic values of content QoS level according to the fuzzy inference rules (see Figure 4) and fuzzy membership of content QoS level (see Figure 3)

The given network bandwidth and response time meet rule 4, 5, 7, and 8. For rule 4, $\mu_{Content\_QoS\_level=low} = \min(\mu_{Network\_bandwidth=medium}, \mu_{Response\_time=short})$ =0.25, which means the content QoS level should be "Low" with confidence value of 0.25. Here we use "min" function, because the certainty of the QoS level is the minimum of the certainty of each component in its fuzzy rule. Similarly, for rule 5, $\mu_{Content\_QoS\_level=medium}$ =0.25; for rule 7, $\mu_{Content\_QoS\_level=medium}$=0.6; for rule 8, $\mu_{Content\_QoS\_level=high}$ =0.4.

Step 4: Transfer the linguistic values of content QoS level into a crisp value according to equation 3

$$Content\_QoS\_level = \frac{0.25 \times \frac{1}{6} + 0.25 \times \frac{3}{6} + 0.6 \times \frac{3}{6} + 0.4 \times \frac{4}{6}}{0.25 + 0.25 + 0.6 + 0.4} \approx 0.49$$

Step 5: Decide the final QoS level

According to Figure 3, the content QoS level, 0.49 is between "Low" and "Medium", but mainly belongs to "Medium", because $\mu_{Content\_QoS\_level=medium}$ is much larger than $\mu_{Content\_QoS\_level=low}$. Hence, we decide the final QoS level as "Medium".

5. QoS Mapping

As the computer itself cannot understand QoS values suggested by the fuzzy recommendation, the content QoS levels should be mapped to machine understandable parameters. Unlike our system, other systems usually conduct the QoS mapping statically before applications are started. It does not take into account the changing device features. Sometime the QoS cannot be guaranteed. For instance, the frame size largely relies on the resolution size of the device. It is not sound to determine QoS parameter without considering the device's capabilities. We propose an adaptive QoS mapping strategy, which dynamically sets quality parameters at run-time according to the capabilities of client devices. Therefore the device features are taken into account in provisioning learning content to pervasive devices.

We could map different QoS parameters for different media modalities, e.g. video, audio, image, respectively. In this paper, we take video streaming as example. Assume the QoS dimensions include frame size, format, frame rate, and quantization scale. The frame size is divided into 8 levels, i.e. 740×480, 640×480, 480×360, 360×240, 240×176, 176×144, 160×120, and 128×96. The maximum frame size relies on the display resolution of the device. The format depends on the operating system and playing software installed. Usually the maximum frame rate for video streaming is 30fps. The quantization scale is related to the image quality and takes integer values ranging from 1 to 31. A lower value means better quality but larger files. Using a value of 1 theoretically leads to highest image quality but generates very large files, so in practice we use 2 as the maximum quality quantization value. The 5-level QoS mapping is as follows:

- Q5 (Very high): set the quantization scale as 2, and set the other three categories as the maximum value that the device supports;
- Q4 (High): on the basis of Q5, decrease the frame rate as 20fps and set the quantization scale as 10;
- Q3 (Medium): on the basis of Q4, decrease the frame rate as 15fps, set the quantization scale as 17 and decrease the frame size one level if it is possible;
- Q2 (Low): on the basis of Q3, decrease the frame rate as 10fps, set the quantization scale as 24 and decrease the frame size one level if it is possible;
- Q1 (Very low): on the basis of Q2, decrease the frame rate as 5fps, set the quantization scale as 31 and decrease the frame size one level if it is possible.

Suppose the device's display resolution is 640×480 and the format supported is mpeg4, then the five QoS levels: Q5 (Very high), Q4 (High), Q3 (Medium), Q2 (Low), and Q1 (Very low) are mapped to [*(frame size, 640×480)*, *(format, mpeg4)*, *(frame rate, 30fps)*, *(quantization scale, 2)*], [*(frame size, 640×480)*, *(format, mpeg4)*, *(frame rate, 20fps)*, *(quantization scale, 10)*], [*(frame size, 480×360)*, *(format, mpeg4)*, *(frame rate, 15fps)*, *(quantization scale, 17)*], [*(frame size, 360×240)*, *(format, mpeg4)*, *(frame rate, 10fps)*, *(quantization scale, 24)*], [*(frame size, 240×176)*, *(format, mpeg4)*, *(frame rate, 5fps)*, *(quantization scale, 31)*], respectively.

So given the device capabilities and the suggested content QoS level, the QoS mapping finally decides the presentation form of the learning content.

6. Implementation and Evaluation

6.1. Prototype Implementation

With the proposed fuzzy recommendation and QoS mapping mechanisms, we developed a QoS-aware pervasive learning system. It was developed with Java and Perl. Figure 5 illustrates the prototype architecture. The QoS fuzzy recommendation server and learning content server were deployed in two computers separately, which addresses the bottleneck issue and improves throughput and scalability of the system. The client side includes PC, handheld PC, and PDA. A HTML form was used for the user to input device parameters and QoS requirements. This information was extracted by a local Web mini-server and sent to the QoS fuzzy recommendation server, which made recommendation according to the information and then decided which content should be offered and its detailed QoS parameters. The url of the selected content was returned to the local Web mini-server. The media player (MPlayer) at the client side used the content url to retrieve the content from the learning content server. To meet different terminal capability and network bandwidth, the learning content needs to be converted accordingly. In our current system, we prepared MPEG-4 media content variations with ffmpeg utility and uploaded them to the content server before delivering.

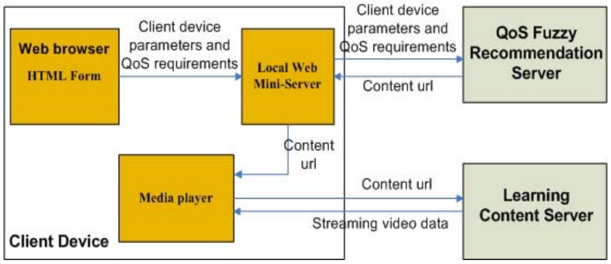
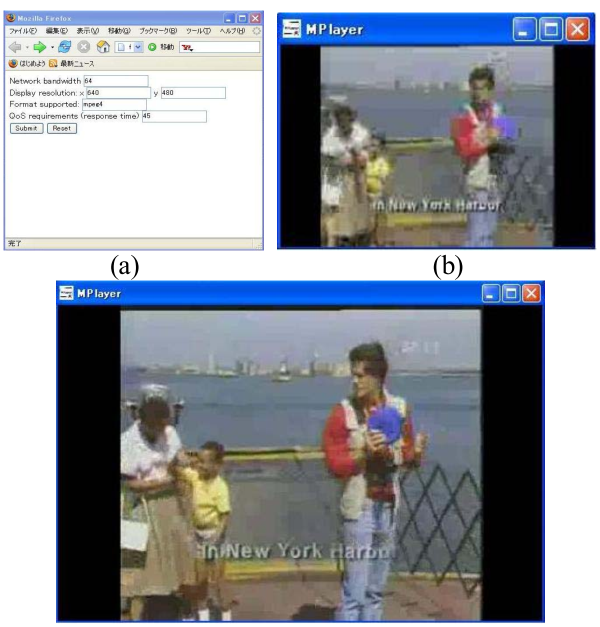


Figure 5. Prototype architecture

Suppose that a learner is currently using a Sony VAIO handheld PC that uses wireless network to connect to the servers. He desires in English learning and decides to access the popular English learning material, *Family Album USA*. The user can explicitly input the device capabilities and the QoS requirements in terms of response time (see Figure 6a). As technical context and QoS requirements changed, the content presentation form varied accordingly. For example, with the same low-bandwidth network, if the user wants to obtain the content very quickly (e.g. 5s), the video playing features 240*176 frame size and 5fps frame rate with very low QoS level (see Figure 6b), while a medium QoS level video with larger size (480*360) and medium frame rate (15fps) displaying when the user is willing to wait for a relative long time e.g. 45s (see Figure 6c).



(a) (b)

(c)

Figure 6. System screenshots: (a) device capability and QoS requirement input; (b) playing very low quality video; (c) playing medium quality video.

6.2. Evaluation

In this section, we present results of our preliminary experiments with the proposed approach. The objective of these experiments is to evaluate the feasibility and user acceptance of the QoS-aware pervasive learning system.

From the technical point of view, we measured the time spent on QoS level recommendation and QoS mapping at the QoS fuzzy recommendation server (Apple MacBook, 2.0 GHz Pentium 4 CPU, 512 MB RAM running Mac OS). It costs a small time, merely about 3ms. So the algorithm is light-weight and feasible to be deployed for content presentation form decision.

We also evaluated the user acceptance of our system from the user's perspective. A user test for 6 participants (four participants are males and 2 are females) was performed. The result is discussed from three aspects: availability, response time, and quality of content. By availability, the service can be accessed through different devices and different networks. It is available anytime and anywhere. Most of the participants are satisfied with the real response time comparing with their time requirement. The quality of content is also acceptable by the subjects with different devices and network conditions.

7. Conclusion

Incorporating QoS-awareness into learning content presentation is crucial for pervasive learning. In this paper, we propose a novel approach to determine appropriate presentation form of the learning content according to user's QoS requirements and device/network capability. The major features of our approach are: (1) combining technical context and user QoS requirements other than merely technical context; (2) adopting fuzzy rules for recommendation rather than crisp rules; and (3) performing dynamic QoS mapping rather than static mapping.

For future work, we plan to integrate the proposed approach into our ULAN (Ubiquitous Learning Architecture for Next generation) project [12] to accomplish QoS-aware educational materials delivery within it.

Acknowledgement

This work is partially supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan under the projects of "Development of Fundamental Software Technologies for Digital Archives" and "Cyber Infrastructure for the Information-explosion Era".

References

[1] A. Schmidt, and C. Winterhalter, "User Context Aware Delivery of E-Learning Material: Approach and Architecture", *Journal of Universal Computer Science* (JUCS), Vol.10, No.1, pp. 28-36, 2004

[2] B. Simon, Z. Mikls, W. Nejdl, M. Sintek, and J. Salvachua, "Smart Space for Learning: A Mediation Infrastructure for Learning Services", *The 12th International Conference on World Wide Web*, Budapest, Hungary, May 2003

[3] T. Lemlouma and N. Layaida, "Encoding Multimedia Presentations for User Preferences and Limited Environments", *Proc. IEEE Int'l Conf. Multimedia and Expo* (ICME 03), IEEE CS Press, 2003, pp. 165-168.

[4] Z. Yu, X. Zhou, D. Zhang, "Supporting Context-Aware Media Recommendations for Smart Phones", *IEEE Pervasive Computing*, Vol. 5, No. 3, July-September 2006, pp. 68-75.

[5] O. Steiger et al., "MPEG-Based Personalized Content Delivery", *IEEE Int'l Conf. Image Processing* (ICIP 03), IEEE CS Press, 2003, pp. 45-48.

[6] L. Zadeh, "Fuzzy Sets as Basis for a Theory of Possibility", *Fuzzy Sets and Systems*, vol.1, 1978, pp. 3-28.

[7] B. Bomsdorf, "Adaptation of Learning Spaces: Supporting Ubiquitous Learning in Higher Distance Education", *Mobile Computing and Ambient Intelligence: The Challenge of Multimedia*, Dagstuhl-Seminar 05181, Saarland, 2005

[8] J. Cao, Na. Xing, A. T.S. Chan, Y. Feng, and B. Jin, "Service Adaptation Using Fuzzy Theory in Context-aware Mobile Computing Middleware", *Proc. 11th IEEE International Conference on Real-Time and Embedded Computing Systems and Applications* (RTCSA'05), Aug. 17 -19, 2005, Hong Kong, pp. 496-501

[9] Takafumi Marutani, Satoshi Nishiguchi, Koh Kakusho and Michihiko Minoh, "Making a lecture contents added deictic information about lecture materials", *The 3rd AEARU Workshop on Network Education*, Seoul, Korea, Dec. 2005.

[10] D. Wichadakul, X. Gu, and K. Nahrstedt, "A Programming Framework for Quality-Aware Ubiquitous Multimedia Applications", *ACM Multimedia 2002*, pp. 631-640

[11] B. Li, and K. Nahrstedt, "A control-based middleware framework for quality-of-service adaptations", *IEEE Journal on Selected Areas in Communications*, Vol.17, No.9, 1999, pp. 1632-1650

[12] ULAN homepage, http://www.ulan.jp/

0-7695-2846-5/07 $20.00 © 2007 IEEE