An Inference Engine for Personalized Content Adaptation in Heterogeneous Mobile Environment*

Seunghwa Lee, Jee-Hyong Lee, and Eunseok Lee**

School of Information and Communication Engineering, Sungkyunkwan University
300 Chunchun Jangahn Suwon, 440-746, Korea
{shlee, jhlee, eslee}@ece.skku.ac.kr

Abstract. In order to overcome the various constraints of wireless environments and provide content according to device specifications and user preference, research relating to content adaptation is gaining in significance. For content adaptation, existing research either prepares content in advance, a reflection of client types which may have access to server, or describes the adaptation rules for dynamic content conversion. However, these require a lot of effort from the content author or system developer, and prospecting the appearance of a new device is a difficult work in today's rapidly changing computing environment. This paper proposes an intelligent adaptation system that automatically extends adaptation rules. The system classifies users into basic categories, then dynamically converts content according to the rule mapping category, offering this result to the user. Then, the system monitors the user action, and performs learning based on this feedback. Moreover, the system has characteristics of offering more personalized content as well as reducing the response time due to reuse of the content generated by same group category. A prototype was implemented in order to evaluate the proposed system in terms of system maintainability, by automatic rule extension, correctness of generated rules, and response time. The effectiveness of the system is confirmed through the results.

1 Introduction

Wireless Internet and various handheld devices are becoming increasingly widespread, due to the rapid development in Information Technologies. Information accessing in mobile environments is gaining in popularity, multimedia content such as Digital Multimedia Broadcasting, and E-learning are now possible. The capacity of handheld devices is continually increasing and new devices of various types are being developed. However, in order to reduce its size and weight, these devices still suffer from various limitations such as small display size, relatively low computing power, and limited battery.

* This work was supported in parts by *Ubiquitous Autonomic Computing and Network Project*, 21th Century Frontier R&D Program, MIC, Korea, ITRC IITA-2005-(C1090-0501-0019), Grant No. R01-2006-000-10954-0, *Basic Research Program* of the Korea Science & Engineering Foundation, and the *Post-BK21 Project.*

** Corresponding author.

H.Y. Youn, M. Kim, and H. Morikawa (Eds.): UCS 2006, LNCS 4239, pp. 158 – 170, 2006.

Thus, research relating to an adaptation system is currently becoming an important issue, in order to overcome various constraints in mobile environments and offer web content according to the characteristics of device and preference of users. The cleanest method to customize the web contents is having content for every possible version, on the server. However, this method requires much effort by the content author, and can incur a heavy storage burden. In addition, prospecting or corresponding with the appearance of a new device is difficult in current rapidly changing computing environments. Hence, adaptation is better performed dynamically, based on understanding the context of the user. Likewise, such dynamic adaptation must build adaptation rules in various client situations. The preparing of adaptation services in advance and the characteristics of new devices is extremely difficult.

This paper proposes an intelligent adaptation system. The system classifies users according to basic classification category, and dynamically converts content using an adaptation rule mapped to the category. Then, it is offered to the user and the system extends group categories and adaptation rules through user feedback. Therefore, users can accept suitable content depending on the situation. Moreover, the proposed system has other characteristic which reuse the adapted contents in order to reduce the time required for adaptation and consumption of computation resources.

A prototype is built in order to evaluate the viability of the proposed approach in terms of system maintainability, automatic rule extension, suitability of generated rules, and required time for adaptation.

The paper is organized as follows: Section 2 introduces related work for the context-adaptive system. Section 3 describes the overall structure and behaviors of the proposed system, designed in order to cope with the weak points of related work. Section 4 describes prototype implementation of the proposed approach and the various evaluation results of using this approach. Finally, Section 5 concludes the paper and identifies possible future work.

2 Related Work

In research relating to adaptation, overcoming the various constraints in a mobile environment, is currently the focus of considerable research in many laboratories. This adaptation can be classified into server side, intermediate, and client side, depending on the location of the adaptation being performed [1].

First, server side adaptation is a method of performing adaptation on the server. A representative study is CARMEN of P. Bellavista et al. [2] and research of A. Pashtan et al. [3]. This method has the advantage is that it can create very suitable content, because adaptation is performed by the author who has a great understanding of the meaning of the content. This method can also be classified *multiple authoring,* which creates a different version in advance, and *single authoring* which is dynamically creates a new version, according to user context, using the single version [4].

Although the multiple authoring method can create a very suitable version, which applies the intent of the author, such accomplishment is actually difficult due to the increased workload of the author. Unlike, the single authoring method, this can reduce the workload of author because it dynamically converts the single version in advance, using rules created by the expert. However, if this conversion is performed on the server, it will increase the response time because it increases the computation burden on the server.

Second, intermediate side adaptation is method for performing adaptation between server and client. This typically refers to a proxy server, it has effectiveness in distributing the computation load. In addition, this method is applied in many existing work such as WebSphere Transcoding Publisher of IBM [5] and research of T. Laakko et al. [6], because it has more characteristics as an *adapter,* even if the server does not offer adaptation functions. However, it suffers from the weak point where it can generate an incorrect result if it is not offered detailed metadata about the contents from the server.

Third, client side adaptation is a method performing adaptation in the client device. A representative research is research of D. Billsus et al. [7]. This method can solve issues relating to privacy which transmit the attributes of the device or data of users externally. In addition, it can embed a function, effectively coping with dynamic resource variation of client devices. However, generally, the client side means a portable device with poor computation power, thus the converting of web content is hard work, and when web content arrives at the client, the conversion contents may not be efficient, because the size of content is not reduced. Hence, this method is used to reconfigure internal components, or for partial content adaptation relating to 'display' via an embedded *stylesheet* [8] or function offered by the browser.

In most existing research, the client transmits context information with a request message when the client requests information from the web server. Then, the server or intermediate side dynamically converts content according to the context of the client. The adaptation rules must be constructed be considering various client situations. However, it is challenging and requires much effort of the content author or system developer. The goal of the proposed system is solving these problems, thus the system has a characteristics which automatically extends adaptation rules using user feedback. Moreover, the research to improve low response time due to adaptation work is in progress, using methods which stores/reuses adapted contents in the cache [9][10]. The proposed system is designed to offer more suitable contents in accordance with integrating a function using the cache which is reusing used contents by users classified with similar characteristics. In addition, the system is designed to structure adaptation processing is distributed between the server and the intermediate side, to make up for the weak points of existing systems previously mentioned.

In the next section, the proposed system designed for efficiently adaptation is introduced in detail.

3 Proposed System

The proposed system is aimed at automatically extending adaptation rules to dynamically convert the content, in order to cope with existing difficult problems, which generate rules by considering various situations. Moreover, the system improves the low response time, due to dynamic content adaptation through reusing the used contents by other users classified as similar characteristics.

3.1 The Proposed System Architecture

The proposed system consist of three parts, Client Module (CM), Intermediate Server Module (ISM), and Server Module (SM). The overall structure is presented in Figure 1. Each module is discussed in greater detail in Section 3.2, the overall behavior is as follows.

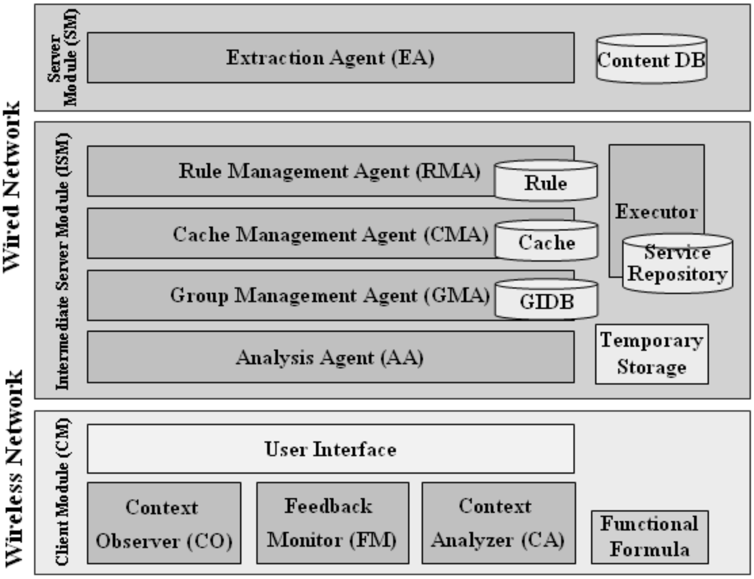


Fig. 1. Proposed system architecture

First, when the client request content, the CM transmits profile information, which is converted with this request message to the ISM. The ISM classifies the user to the most similar category using this information, and searches adapted contents for request in the cache. If an appropriate content should exist, it would be immediately transmitted to the user. Otherwise, the ISM forwards the request message to the original destination server.

The SM extracts the required portion from origin content, and rebuilds the structure of the content as layout. Then, origin image and video, which are embedded in content, are transmitted to the ISM. The ISM receives it, and then performs adaptation according to the adaptation rules mapped to the category. The ISM transmits the adapted contents to the client, and stores it in the cache.

After this, the CM monitors the user's action. If a user should use the adapted content as it is, this is transmitted to the ISM and regarded as positive feedback. The ISM adds the score to the equivalent adaptation rule based on this feedback. However, if users adjust the intensity of service via adjusting the option supported by the application, this is transmitted to the ISM and regarded as negative feedback. The ISM reduces the score of the corresponding adaptation rule, based on the feedback message, and adjusts the position of nodes in the group category. Therefore, the nodes are continually distributed by feedback, and re-clustering is performed. Thus, classification categories are gradually fractioned, and the adaptation rules are automatically extended accordingly. A user accessing the system applies a more specific classification criterion.

3.2 System Components

(1) Client Module

- *Context Observer (CO)*: It gathers the various context information of the user and generates a well-formed document using RDF [11]. In order to reduce the computation burden, the renewal period for static context representing the capability of device and dynamic context such as resource variation are different.

The type of available context is presented in Table.1, and can be extended according to the type of adaptation services. The prototype of the proposed system uses partial context as presented in Figure 2.

Table 1. Type of applicable context

Acquisition method	Context Type	Application
The context information is automatically gathered by the Context Observer (CO)	Acceptable media format of device	Convert the format of original media
	CPU-RAM capacity, current usage	Raise the degrade intensity of service, when usage exceeds the threshold
	Display size	Adjust the size of content into the size of the display
	Acceptable color depth of device	Reduce the color depth of true color media
The context information is explicitly obtained via the user interface	User's preference for each adaptation service	Decide binding of specific services such as *text_to_audio*, when user be exercising or driving
	Preference for language	Parameter of adaptation service as translator
	Preference for color depth	Decide the weight value when convert the color
	(E-learning) Content's degree of difficulty	Form content

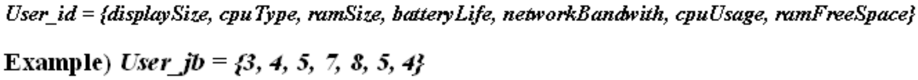
- *Context Analyzer (CA)*: It performs regularization which converts it to the score via comparison with a mapping table. As a result, each context is converted to values between 0 to 1, and is represented in the vector set. The generated vector set is transmitted to the ISM when a user requests content via the interface. Hence, the system can slightly reduce the amount of the transmitted messages.

The dynamic context is computed using each formula, and the static context such as *cpyType*, cannot compute using formulae, is obtained by a mapping table. It is assumed that the formulae for the computation and the mapping table are updated by experts.

$$ramFreeSpace = Rnd\left(\frac{ramainedMemory}{totalMemory} \times 10\right) \tag{1}$$

* In this operation, the *Rnd* mean that a functional symbol for rounding.

In related work, the method to inform the state of client to server side uses the HTTP header [12], or CC/PP [13] information, which is described in the RDF. The server can use the information in order to grasp the characteristics of the client. However, the former suffers from the disadvantage that it is insufficient offered information from the server. The latter makes the amount of messages between client and server slightly increase, although the profile repository is used on the network if the size of the profile is long or just refers to changed parts. For reduction, the proposed system transmits the score as in Figure 2.



User_id = {displaySize, cpuType, ramSize, batteryLife, networkBandwith, cpuUsage, ramFreeSpace}

Example) *User_jb = {3, 4, 5, 7, 8, 5, 4}*

Fig. 2. Example of generated score to represent the user context

- *Feedback Monitor (FM)*: It is embedded in the client device, monitors the user action for adapted contents. The monitoring is accomplished by monitoring the key input of the interface. The essential keys in the feedback message are presented in Figure 3.

```
MessageType = feedback
User_id = 203.252.53.156
ContextScore = {3, 4, 5, 7, 8, 5, 4}
SenderAppType = ClientModule
InterfaceVer = 1.0
FContent = {(image_size_adjust, -10)}
```

Fig. 3. Essential elements of feedback message

The *MessageType* is used to identify the common and feedback messages, and the *User_id* uses the IP address to identify the user. The *FContent* represents the content of the feedback. If user uses the adapted contents as it is, this is regarded as positive feedback, the message is attached as ‘positive’. If a user controls the key relating to the adaptation service offered from the interface, this is regarded as negative feedback, this information is concretely attached to *FContent* key.

(2) Intermediate Server Module

- *Analysis Agent (AA)*: It identifies the type (i.e., Request, Feedback) of message from client. For the request, the request and profile component are detached by message parsing

- *Group Management Agent (GMA)*: It manages the group category. Firstly, it compares similarity between the user context set and group category. The similarity is applied with the extended Euclidean distance [14], which is used to compare the similarity of two vectors. The applied formula is as follows.

User Context Set: $U = (u_1, u_2, u_3 \cdots u_n)$, *Group Category:* $G_i = (g_1, g_2, g_3 \cdots g_n)$

* where, i : the index of group categories, $i \geq 1$.

$$app(U, G_i) = sim_{G_i \in T} \sqrt{\sum_{n=1}^{m} (u_n - g_n)^2} \tag{2}$$

* where, $app(U, G_i)$: measurement of the similarity between user profile and group category

$sim_{G \in T}$: each group category included in total category set T

m : the number of context information

If similarity result is duplicated, the duplicated values are computed as shown in formula (3), by assigning weights from the first context.

$$\begin{array}{c|l} & w, \quad w-1, w-2 \cdots \\ \hline G_1 & g_{11}, g_{12}, g_{13}, \cdots g_{1n} \\ G_2 & g_{21}, g_{22}, g_{23}, \cdots g_{2n} \\ \vdots & \qquad \vdots \\ G_k & g_{k1}, g_{k2}, g_{k3}, \cdots g_{kn} \end{array} \tag{3}$$

* where, k: the number of group categories which have duplicated result. $k \le |T|$

w: weight factor. $w = |n|$

n is the number of context factors

- *Cache Management Agent (CMA)*: This agent searches adapted contents in the cache, generated from group and classified in the cache as similar to the user. If appropriate content should exist, it would be immediately transmitted to the user. Otherwise, the request is forwarded to the original destination, and it is informed to the RMA.

Table 2. Table to manage the cache

Group type	Resource ID	Applied Service list
3334444	http://selab.skku.ac.kr/test1.html	(image_size_converter, 30), (image_degrader, …
5554444	http://selab.skku.ac.kr/test2.html	(image_size_converter, 10), (image_degrader, …
3334444	http://jbmania.netcci.org/sample..	(video_size_converter, 15), (frame_rate_degr …
:	:	:

- *Rule Management Agent (RMA)*: This agent searches the list of adaptation service mapping with the group category in the rule database.

- *Executor*: It converts the received media source from the SM, based on the adaptation list. The components for adaptation exist in the service repository. The executor uses it by calling with parameters as intensity. The adapted content is transmitted to the user, and is stored concurrently to the cache by the CMA. In addition, the executor temporary stores adaptation information such as user id, group category, content URL, adapted service list, and its intensity, in order to analyze feedback.

(3) Server Module

- *Extraction Agent (EA)*: It generates the new content with extraction of required elements from the source file in the original content DB, and decides on the content layout using a *stylesheet* and transmits the original video or audio which will be included in the document to the ISM. The transmission of the original content may result in network traffic increasing, however, the proposed system is designed to preferentially distribute the server's workload. However, other research is in progress, the workload is dynamically distributed by reflecting each state of resources among server, proxy, and neighboring proxy.

3.3 Extension of Rule Using Feedback

The FM embedded in the client device, monitors the application control of users about the adapted content, and informs this to the ISM. Monitoring can be accomplished by analyzing the inputted key values. The AA informs the feedback message to the GMA with the stored information in temporary storage. If the message is negative feedback, the GMA moves the user node to the intensity as presented Figure 4.

Then, the nodes are continually distributed by feedback, and system re-clustering is performed via the k-means algorithm. Hence, group categories fractionize. A specification for the appropriate number of categories is required.

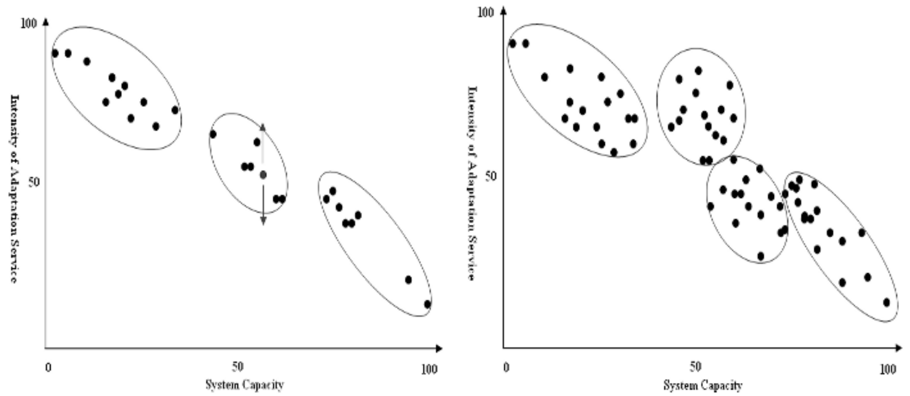


Fig. 4. Movement of nodes in the feedback and re-clustering phase

The RMA inherits adaptation rules of the previous category in order to generate the adaptation rule mapping with the new generated category. The new adaptation rules are then generated by adjusting the intensity of adaptation. This method is not customization for the specific user, but it is suited to similar majority users. However, eventually the result reflects the preference of real users unforeseen by the developer, and user satisfaction increases.

The rules managed by RMA, are presented as Table.2. In this table, all service lists and the score for each service are omitted. If positive feedback is detected, the score of corresponding rules increases. Through this score, generated groups and rules from

Table 2. Table to manage adaptation rules

	Group type	Image	video
	3334444	(image_size converter, 30), (image_degrader,…)	(video_size_converter, 15), …
	5554444	(image_size converter, 10), (image_degrader,…)	(video_size_converter, 10), …
	7774444	(image_size converter, 5), (image_degrader,…)	(video_size_converter, 5), …
→	3457854	(image_size converter, 25), (image_degrader,…)	(video_size_converter, 15), …

irrelevant feedback are automatically removed after a period of time, and adaptation services classed as acceptable by the majority of users are maintained.

4 Experiments and System Evaluation

In this section, the proposed approach is evaluated in terms of system maintenance by automatic rule generation, suitability of generated rules, and the required time for adaptation. The performance of the system was measured, based on prototype implementation. The specification of the devices used for the implementation of prototype is as follows. Firstly, the SM and ISM were implemented on desktop PCs equipped with an 3GHz Intel Pentium-4 processor, 1Gbyte RAM, and connected over a wired LAN which has a speed between 14Mbps and 28Mbps. The client device used a desktop pc with the same specifications at above, notebook pc, and HP 5500 PDA which has a display size of 240 * 320, a 400MHz CPU, and 128M RAM.

The module for PDA is implemented by Embedded Visual C++, and the another modules are implemented in Java with JADE and JADE-LEAP [15]. In addition, the adaptation service used in evaluation used the image_converter, which was implemented in Jimi [16] and Java Advanced Imaging (JAI) [17]. This service is an adaptation service which takes parameters such as filename, width, height, and quality then handles the image. In addition, XML related technologies such as DOM and XPath were used to handle web documents.

(1) Maintenance by automatically rule generation
The first test was to confirm that the group categories and the adaptation rules (intensity) are automatically extended by feedback.
The three basic categories are created in the GIDB, and the adaptation rule and intensity are mapped to these categories, Multimedia content is made as the target content such as relatively large image files required for adaptation, and documents.

- image1.gif: display_size 2264 * 1448, file_size: 418kbytes
- image2.gif: display_size 1948 * 1216, file_size: 167kbytes
- :

We made one group which is composed of 10 users having different device, access and download to these content. This result, all users suffer from negative feedback, adjusting the option of the application because categories are insufficient. Hence, the classification category is increased, and the adaptation rule is extended according to the adjusting of option. Then, when the same group requests content again, it can be

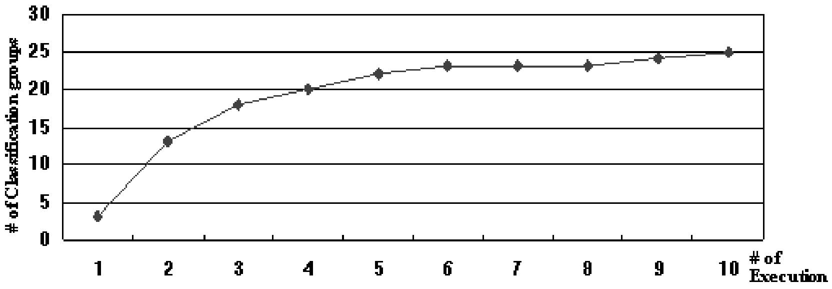


Fig. 5. Number of classification categories extended

known that the number of classification categories was increasing to small range because the categories subdivide.

As a result of the repeat of this process, the increase range is approximately the number of categories, and is therefore relatively low, flattening when reaching higher values such as 20. Through this result, it is confirmed that the number of categories according to the number of users, has not increased to infinity. In addition, the number of categories may not increase by irrelevant feedback because developers define the number of categories and manage this value via the score. This result removes concern for the system, which is massively increasing the number of categories, and indirectly verifies the system can cope with new devices emerge. Moreover, the suitableness of rules is confirmed through the result of decreased of negative feedback.

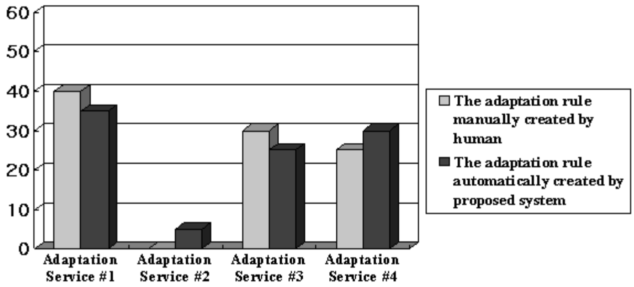


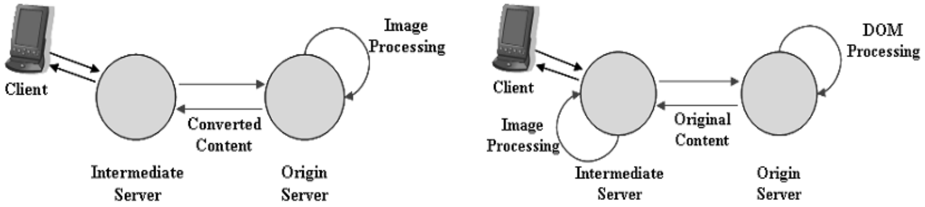
Fig. 6. Evaluation result for fitness of rules generated by the system

(2) Suitableness of automatically extended rules

Secondly, the adaptation rules are generated manually for 10 clients, which will access to the system in order to evaluate the suitability of automatically extended adaptation rules. These rules and the adaptation service rules generated by the system are compared. This comparison is possible due to the intensity of adaptation service rules represented as a numerical value. In the results, the required parts are almost consistent, it can be confirmed that automatically generated rules by feedback are suitable. The evaluation result is presented, as presented in Figure 6.

(3) The reduction of response time

The third test is to evaluate the structural advantage of the proposed system and the advantages of using the cache. The same request is made to the proposed system and existing server side adaptation system, and the response time for these is compared.



(a) Existing server side adaptation

(b) The proposed system

Fig. 7. Comparison of existing server side adaptation with the proposed system which is distributed processing of the adaptation job

The server side adaptation is a method where main adaptation work is processed in the server side as Figure 7-a. This is in contrast to the proposed system, which has structure as Figure 7.b, that the adaptation work relating to composition of content is processed on the server side; the converting work consumes considerable resources, and is processed in intermediate server.

The content from proxy to server is requested, and the response time is measured, without the request part of the client which is unnecessary in this test. In this test, a jpg image file having a 680 * 500pixel, and 177kbytes size is used. It is converted to 400 * 300 pixel, and quality is degraded 50%. The converted image is decreased to 19kbytes. The response time for these files presents average results as follows:

- The time for image processing: 1300 ms
- The time for transmitting the adapted content (19kbytes) from server to proxy (existing method): 30 ms
- The time for transmitting the original content (177kbytes) from server to proxy (proposed method): 40 ms

In the test result, it can be known that the content before adapting and the content after adapting do not have much difference in transmission in a wired environment. In ordinary situations, performing all adaptation work on the server side has more efficient results. In addition, generally, the web server has higher performance than the proxy, but, in this test, we use the systems having identical specification.

However, when a considerable workload occurs in the server, the result is very different. The *process_generator* generating the dummy process in the server is executed in order to simulate situations where the number of users connecting to the server is increasing. This developed application uses the log which records the number of accessing users and the usage of resources on the server, in order to simulate the increasing number of users. In test processes, the response time of existing server side adaptation system increases sharply according to increasing workload, unlike the

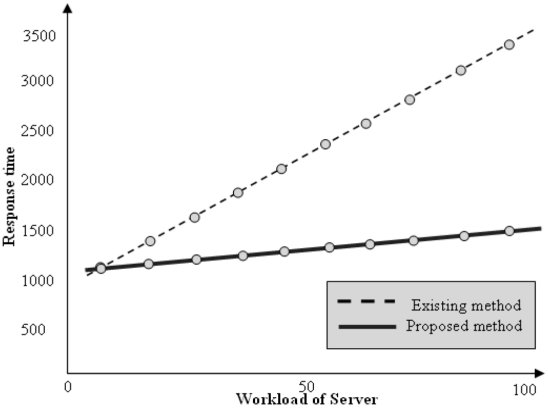


Fig. 8. Comparison of response time along with workload of the server

proposed system, which maintains consistency. The evaluation result is presented in Figure 8.

Through this result, the efficiency of the proposed system in terms of distributing adaptation work is confirmed. In addition, another result is verified, where response time is reduced through reusing content.

The quantitative measurement of user satisfaction is omitted due to the difficulty of determining this value. It can be known that more suitable content is offered due reusing the adapted contents generated by the same category.

5 Conclusion

In existing work, in order to offer the content adaptation service, the creation of multiple content forms, or rules for dynamic adaptation are made in advance, by considering the device characteristics of various users. However, this process is extremely difficult. The proposed system automatically extends adaptation rules depending on user feedback, in order to solve this problem.

A prototype was implemented in order to evaluate the proposed system in terms of fitness of automatically extended rules and system efficiency. Through the result, negative feedback decreases, and adaptation rules are automatically extended to a positive form for user satisfaction, and it is verified that automatic adaptation to a new device is possible. Moreover, by comparing with the server adaptation system, the structural efficiency of the proposed system is evaluated, which distributes the processing of adaptation work, in order to avoid excessive workload on one system.

As future work, the public owned function for information among the ISMs will be integrated, in order to solve the ‘learning speed’ problem of the system. In the future, a new adaptation strategy mechanism will be generated, based on the meaning of content, in order to realize increasingly suitable adaptation mechanisms.

References

1. Margaritis Margaritidis and George C. Polyzos, "Adaptation techniques for Ubiquitous Internet multimedia", Wireless Communication and Mobile Computing, vol.1, no.2, pp.141-163, Jan.2001
2. Paolo Bellavista, Antonio Corradi, Rebecca Montanari, and Cesare Stefanelli, "Context-Aware Middleware for Resource Management in the Wireless Internet", IEEE Trans. on Software Engineering, vol.29, no.12, Dec.2003
3. Ariel Pashtan, Shriram Kollipara, and Michael Pearce, "Adapting Content for Wireless Web Services", IEEE Internet computing, vol.7, no.5, pp.79-85, Sep.2003
4. Rotan Hanrahan, Roland Merrick, Candy Wong, Michael Wasmund, Rhys Lewis, and Tayeb Lemlouma, "Authoring Techniques for Device Independence", World Wide Web Consortium, Note, NOTE-di-atdi-20040218, Feb.2004
5. IBM WebSphere® Transcoding Publisher, http://www-306.ibm.com/software/pervasive/transcoding_publisher
6. Timo Laakko and Tapio Hiltunen, "Adapting Web Content to Mobile User Agents", IEEE Internet computing, vol.9, no.2, pp.46-53, Mar.2005
7. Daniel Billsus, Clifford A. Brunk, Craig Evans, Brian Gladish, and Michael Pazzani, "Adaptive Interfaces for Ubiquitous Web Access", Communication of the ACM, Vol.45, No.5, pp.34-38, May.2002
8. Mark Butler, Fabio Giannetti, Roger Gimson, and Tony Wiley, "Device Independence and the Web", IEEE Internet Computing, Vol.6, No.5, pp.81-86, Sep.2002
9. Cheng-Yue Chang and Ming-Syan Chen, "On Exploring Aggregate Effect for Efficient Cache Replacement in Transcoding Proxies", IEEE Trans. Parallel and Distributed systems, vol.14, no.6, pp.611-624, Jun.2003
10. Claudia Canali, Valeria Cardellini, Michele Colajanni, Riccardo Lancellotti, and Philip S. Yu, "A two-level distributed architecture for efficient Web content adaptation and delivery", Proc. SAINT 2005, pp.132-139
11. W3C – Resource Description Framework (RDF), http://www.w3.org/RDF/
12. W3C – Hypertext Transfer Protocol (HTTP 1.0), http://www.w3.org/protocols/http/
13. W3C - Composite Capability/Preference Profiles (CC/PP), http://www.w3.org/Mobile/
14. Michael J. A. Berry, Gordon S. Linoff, "Mastering Data Mining: The Art and Science of Customer Relationship Management, Wiley, 2000
15. Telecom Italia Lab - Java Agent DEvelopment Framework (JADE), http://jade.tilab.com/
16. Sun Microsystems – Java Image Management Interface API, http://java.sun.com/products/jimi
17. Sun Microsystems – Java Advanced Imaging (JAI) API, http://java.sun.com/products/java-media/jai/
18. Yu Chen, Xing Xie, Wei-Ying Ma, and Hong-Jiang Zhang, "Adapting web pages for small-screen devices", IEEE Internet computing, vol.9, no.1, pp.50-56, Jan.2005
19. Wai Yip Lum and Francis C.M. Lau, "A context-aware decision engine for content adaptation", IEEE Pervasive computing, vol.1, pp.41-49, Jul.2002