Decision Engine for Data Adaptation Service in Advanced Collaborating Environment

Mohammad Rezwanul Huq, Khandoker Tarik-Ul Islam, Young-Koo Lee, Byeong-Soo Jeong, Sungyoung Lee

Computer Engineering Department, Kyung Hee University
Yongin-Si, Gyeonggi-Do, Korea, 446-701
rhuq@oslab.khu.ac.kr, khtarik@oslab.khu.ac.kr, yklee@khu.ac.kr, jeong@khu.ac.kr, sylee@oslab.khu.ac.kr

***Abstract*— Content adaptation is very much necessary for effective and efficient sharing of files. Now-a-days, people are using devices which vary in their configuration. Moreover, each user may have their own preference whenever they want to share files. To address device heterogeneity as well as user preference, we need to adapt the file at the time of sharing. In this paper, we propose a decision engine for file adaptation in advanced collaborating environment. The decision engine selects the best match adaptation settings based on both user preference and device capabilities. Moreover, we discuss about our first prototype implementation here. At it's current stage, our prototype only deals with image files. Our prototype implementation realizes our designed framework and demonstrates the viability of our approach.**

I. Introduction

In collaborative environment, users may want to share files which may differ in their type and size. Therefore, sharing various types of data is a major user requirement. Moreover, adapting the content of the file is required for device heterogeneity. This type of adaptation is based on the target device alone, also known as device-centric approach. Device-centric adaptation only considers the capabilities of a device and adapts data in such a manner which will be renderable in the target device. In this case, the system is most likely to produce the highest quality adaptation. We have to also consider the user preferences while adapting data. The goal of the user-centric approach is to generate an adapted version such that the version itself is a best match to user preferences thus satisfying the user.

In this paper, we propose a decision engine for data adaptation considering user preferences and device capabilities. Our decision engine maintains a balanced approach of data adaptation. We store user preferences on various dimensions of a file type. In our design, we have used cosine similarity measure and Euclidean distance for finding out the best match converter. Capabilities of user devices are also stored to identify the best possible adaptation method which satisfies user preferences as well as device capabilities. User preferences and device capabilities are stored in an ontology-based database. We are developing a prototype implementation based on our design which can be applicable to advanced collaborating environment. The prototype implementation provides the best possible adaptation scheme considering user preferences and device capabilities.

The rest of the paper is organized as follows. Section 2 begins the discussion of related work. In section 3 we discuss about advanced collaborating environment followed by the definition of data adaptation service. The proposed framework for decision engine has been discussed in section 5. Next, our prototype implementation has been included in section 6. Finally, we talk about some future work and conclude in section 7.

II. Related Work

Several research papers have been published for providing effective and efficient data adaptation. The principles of device-independence were addressed in [1], where, for achieving the independence, it should be possible for a user to obtain a functional presentation via any access mechanism. The study in [2] follows the same way and recommends that clients can be viewed as varying along three important dimensions: network, hardware and software. Content adaptation systems can make decisions on the response content versions or adaptation strategies based on some situational context. Some works can be found in [3], [4] mentioned about this trade-off policy. The judgement for this trade-off policy can also be derived using some score based evaluation where users can assign a score to different adaptation scheme to show their preference on the decision of adaptation [5], [6], [7]. But in practical situations, it might be difficult for users to assign a score to each adaptation scheme. To make this assignment automatic, the approach resource-based content value was proposed [5], [6], [7]. In this approach, the content value is dependent upon the resources in the client device that can be used to render the file, not the degree of satisfaction perceived by a user.

In this paper, we propose a framework for decision engine which will provide the best possible adaptation scheme considering user preferences as well as device capabilities. We propose a decision engine which uses cosine similarity to find out the best match adaptation settings based on user preferences and device capabilities. This decision engine has been integrated into data adaptation service so that users can share adapted file according to their preference and device capabilities.

III. Target Environment

Due to the advancement in the field of networking and multimedia technology, The traditional video conferencing becomes obsolete now-a-days. People in different places want to interact with each other in a more convenient and meaningful way. That is why, the concept of advanced collaborating environment is essential to provide interactive communication among a group of users. The ACE aims to bring together the right people and the right data at the right time in order to perform a task, solve a problem, or simply discuss something of common interest [8] [9]. Some early prototypes of ACE have been mainly applied to large-scale distributed meetings, seminar or lectures and collaborative worksessions, tutorials, training etc. The primary intention of ACE is to provide advanced collaboration among users.

One of the major user requirements in advanced collaborating environment is sharing files among users. Files exist in different forms and size. Therefore, for effective and efficient file sharing sometime these files need to be adapted from one form to another. At the time of data adaptation, we need to consider the user preferences and device capabilities.

IV. Data Adaptation Service

The principle concern of data adaptation service is to adapt data according to the user's preferences and device capabilities. User preferences will be stored earlier. For a particular file format, we will identify some quality dimensions. Users will put their preference on different quality dimensions. According to their preference, user preference vector will be formed. We have a set of converters having converter preference vector. Our decision engine chooses the best match converter using cosine similarity between converter preference vector and user preference vector. At the time of actual adaptation we will also consider the target device capabilities.

Figure 1 refers the block diagram of data adaptation service. Here, we will choose the adaptation strategy based on the decision provided by our decision engine. Based on our decision logic the decision engine is supposed to provide the best adaptation scheme for that particular file.

Formally we can define our problem in the following manner.

"In advanced collaborating environment, for effective and efficient file sharing, sometimes we might need to adapt data based on user preferences and device capabilities. Our primary concern is to build a decision engine which will provide us the best possible data adaptation scheme considering all these constraints from user and device point of view."

V. Decision Engine

The decision engine (see figure2) is the central part of data adaptation service. Data adaptation service will adapt data according to the decision provided by the decision engine. Decision engine will take decision based on user preferences and device capabilities. The methodology for selecting appropriate data adaptation scheme is as follows.

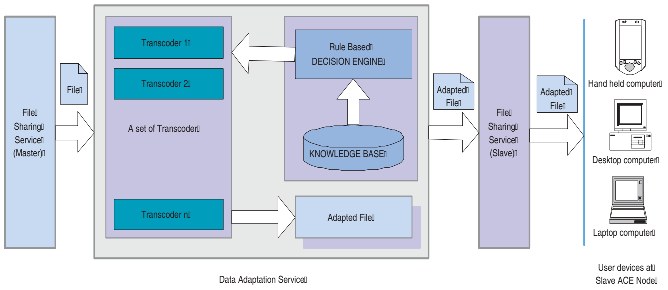

Fig. 1. Block Diagram of Data Adaptation Service

- Decision engine chooses the best possible adaptation scheme based on user preferences.
- Then it checks the device capabilities of the target device.
- If device capabilities support the current adaptation scheme then corresponding transcoder will be selected for data adaptation otherwise another adaptation scheme will be chosen based on user preferences.

Next, we will describe about the procedure of choosing the best possible adaptation scheme based on user preference as well as our overall decision process.

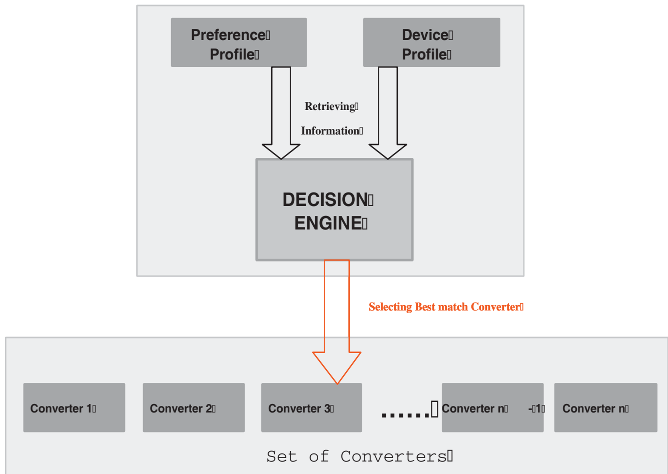

Fig. 2. Block Diagram of Decision Engine

A. Definition of Related Terminologies

Decision Engine provides the best match converter for a given file considering the user preferences and device capabilities. We use some terminologies related to Decision Engine for decision generating mechanism.

Definition 1-Quality Dimension: It can be defined as a set of properties for a particular file format. These quality dimensions together forms the QoS for any content type. As for example $(QoS_{imagefile})$ = (color, scaling, sharpness)

Definition 2-Quantization Step: It defines the scale that is applicable to a particular quality dimension. Take image as an example, the scale has 1-bit, 8-bit, 24-bit and so on, as quantization steps.

Definition 3-Quality Value: It can be defined as a numerical value based on the quantization step. The quality value will be computed by quality value modeling function described later. The quality value ranges from 0 to 1.

Next, we need the modeling functions f that capture the behavior of qv against the variation of qs. It is denoted by $qv = f_i(qs)$ for the ith quality domain.

We can use both of the first order and second order modeling to monitor the change of qv against qs. The first order quality value modeling function is as follows.

$qv = (qs - qs_{min})/(qs_{max} - qs_{min})$ (increasing)

where qs_{max} is the maximum step possible in this quality dimension (e.g. 100 percent scaling factor), qs_{min} is the lowest step possible (e.g. 1 or 2 percent scaling factor).

For the second order modeling, the modeling curve is characterized by the second order equation.

$qv = a * qs^2 + b * qs + c$

Definition 4-User Preference Value (Upv): User Preference Value for a particular quality dimension represents the preference of user for that quality dimension. The higher the value, the more the preference. The range for user preference value is from 0.0 to 1.0. The scale is 1 to 10 so we have divided the raw value by 10. As for example,

$Upv_{color} = 10/10 = 1.0$
$Upv_{scaling} = 5/10 = 0.5$
$Upv_{sharpness} = 5/10 = 0.5$

Definition 5-User Preference Vector (Vup): User preference Vector used to represent the normalized user preference in a vector. As for example,

$Vup = (Upv_{color}, Upv_{scaling}, Upv_{sharpness})$
$Vup = (1.0, 0.5, 0.5)$

Definition 6-Converter Preference Value (Cpv): Converter Preference Value for a particular quality dimension represents the preference of converter settings for that quality dimension. The higher the value, the more similar to the original file.

Converter preference value comes from the quality values for a particular quality dimension. For a particular converter, Cpv_{color} can be any value from the set of quality values of color dimension. One thing needs to be remembered that, two converters cannot have the same set of converter preference values for all dimensions.

Definition 7-Converter Preference Vector (Vcp): Converter Preference Vector used to represent converter preference value in a vector.

Definition 8-Similarity (SIM): Similarity defines the similarity measure used to identify the similarity between User Preference Vector and Converter Preference Vector.

We use cosine similarity measure for this purpose. Cosine similarity measure finds the angle between two vectors. If the value becomes 1, it means that two vector coincides with each other. If the value is 0, it means that two vectors are perpendicular to each other.

Definition 9-Distance (DIST): Distance is a measure used to calculate the distance between converter preference vector Vcp and user preference vector Vup in n-dimensional space where, n is the number of quality dimensions.

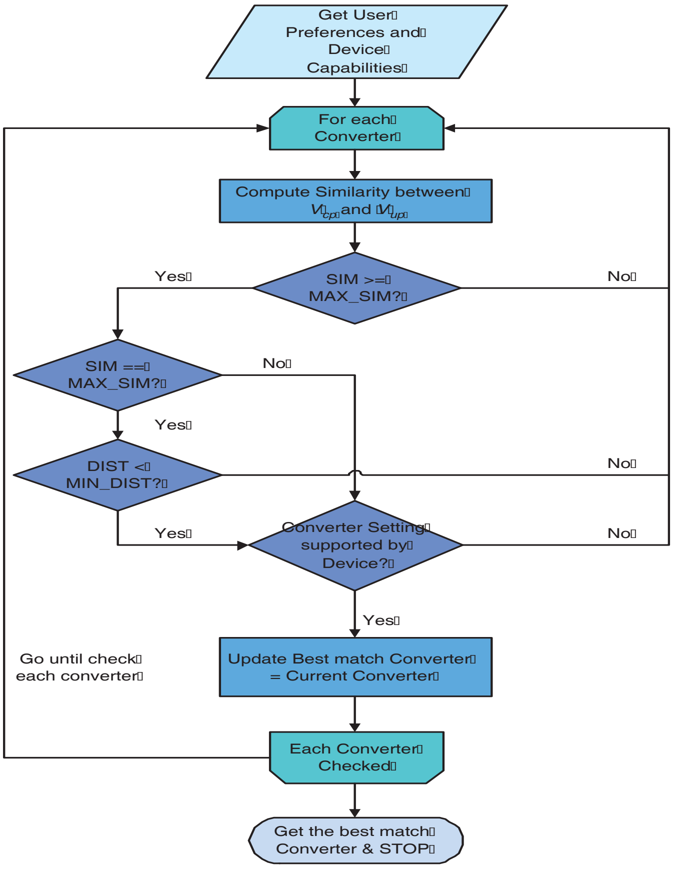

Fig. 3. Flow Chart of Decision Logic

We use DIST measure when multiple number of converters having same similarity to user preference. The converter preference vector having smallest distance from user preference vector will be chosen as the best match converter.

B. Decision Logic

At first, Decision Engine retrieves all necessary information including user preferences, device capabilities. Then, it searched for the best match converter. Currently, we follow linear searching. So, for each converter preference vector, decision engine will calculate the similarity between user preference vector and that particular converter preference vector. Decision Engine chooses the converter having highest similarity with user preference. If multiple converters have same similarity, decision engine selects the converter preference vector having smallest distance from the user preference vector.

At the time of choosing best match converter decision engine also looks for whether this converter settings can be supported by device capabilities or not. If converter settings are not supported by user device, decision engine rejects the converter even though the rejected converter might have the highest similarity.

Decision Engine chooses the converter having highest similarity with the user preferences as well as the converter settings

```
Input: V_up, {V_cp}, Device_color_depth, Device_screen_resolution, Original_Image_Resolution

Output: Best match V_cp

FOR each V_cp in {V_cp}
    COMPUTE SIM
    COMPARE SIM >= MAX_SIM
        IF TRUE, COMPARE SIM = MAX_SIM
            IF TRUE, COMPARE DIST < MIN_DIST
                IF TRUE, COMPARE Device_color_depth supports N_cp-color
                    IF TRUE, MIN_DIST = DIST and Best match V_cp = V_cp
                        BREAK
                    ELSE, CONTINUE
                ELSE, CONTINUE
            ELSE, COMPARE Device_color_depth supports N_cp-color
                IF TRUE, MAX_SIM = SIM and MIN_DIST = DIST and Best match V_cp = V_cp
                    BREAK
                ELSE, CONTINUE
        ELSE, CONTINUE
```

Fig. 4. Algorithm for Decision Engine

can be supported by the user device. This is how, our decision algorithm provides the best match converter considering both of the user preferences as well as device capabilities. Figure 3 and 4 shows the flow chart for our decision logic and formal decision algorithm respectively.

VI. Prototype Implementation

We have implemented our decision engine using Python[10]. Initially, our prototype implementation can only adapt the image files. For adapting image files, we use PIL[11] (Python imaging library). PIL will facilitate the adaptation procedure for a particular transcoder. As for example, if decision engine provides that 24-bit image should be adapted to 1-bit image, using PIL we can adapt the original image.

Moreover, we are using Oracle 10g for storing user preferences, device capabilities. This information will be fetched at the time of choosing a transcoder by decision engine. Here, we use ontology-based database. We have installed cxOracle[12]library which gives the interface between oracle and python.

We have also provided the user interface to put their preferences on different quality dimensions as well as their device capabilities. This user interface has been designed in wxPython which is a graphical user interface for python.

At the beginning, the user enters his/her e-mail address and preferred file type for sharing. Then, he/she presses the appropriate button for connecting Master ACE node. After pressing the connect button the master file sharing service will connect to the venue data store and retrieves file of user mentioned type. Then, the user will select one of the files. The specified file will be downloaded through the master file sharing service, the decision engine takes the decision of selecting appropriate adaptation method and then it will be passed to the data adaptation service. The function DataAdapter() takes the file and convert it based on the decision provided by the decision engine.

Then adapted file will be sent to Slave ACE node by master file sharing service and the adapted file will be stored into the local storage of slave ACE node.

VII. Conclusion and Future Work

In this paper, we proposed the framework of decision engine for data adaptation service in advanced collaborating environment. File adaptation is very much necessary because of the differences in user preferences and device capabilities . Our main contribution is to integrate data adaptation service into the ACE environment as well as propose a balanced approach of adapting data. We have considered user-centric and device-centric approach for data adaptation and devise a decision logic which follow the hybrid approach to provide the best possible adaptation scheme. We have already developed our decision engine module and integrated with the overall framework. Initially, our primary concern is to adapt image files only. Later, we will try to adapt all kinds of files using same decision engine. This paper proves the necessity of data adaptation service for ACE environment and also provides a framework for realizing this service based on decision engine.

Acknowledgement

This work is supported by Ubiquitous Computing and Network (UCN) grants Z1500-0603-0004, the Ministry of Information and Communication (MIC) 21th Century Frontier R&D Program in Korea.

References

[1] *Device Independence Principle (http://www.w3.org/TR/2001/WD-di-princ-20010918/)*, W3C.

[2] A. Fox and E. Brewer, "Reducing www latency and bandwidth requirements by real-time distillation," in *Proc. of Fifth Intl World Wide Web Conf. (WWW5)*, May 1996.

[3] J. Chen, Y. Yang, and H. Zhang, "An adaptive web content delivery system," in *Proc. of Intl Conf. Adaptive Hypermedia and Adaptive Web-Based Systems (AH2000)*, August 2000.

[4] W. Ma and et. al, "A framework for adaptive content delivery in heterogeneous network environments," in *Proc. Conf. Multimedia Computing and Networking*, January 2000.

[5] R. Mohan, J. Smith, and C. Li, "Adapting multimedia internet content for universal access," *IEEE Trans. Multimedia*, vol. 1, no. 1, March 1999.

[6] C. Li, R. Mohan, and J. Smith, "Multimedia content description in the infopyramid," in *Proc. of IEEE Intl Conf. Acoustics Speech and Signal Processing (ICASSP 98)*, May 1998.

[7] J. Smith, R. Mohan, and C. Li, "Transcoding internet content for heterogeneous client devices," in *Proc. IEEE Intl Symp. Circuits and Systems (ISCAS)*, May 1998.

[8] B. Corri, S. Marsh, and S. Noel, "Towards quality of experience in advanced collaborative environments," in *Proc. of the 3rd Annual Workshop on Advanced Collaborative Environments*, 2003.

[9] S. Han and J. Kim, "Adaptive high-quality video service for network-based multi-party collaboration," *Proc. of SPIE*, vol. 6015, October 2005.

[10] *Python Programming Language*, python.org.

[11] *Python Imaging Library (PIL) (http://www.pythonware.com/products/pil/)*, pythonware.com.

[12] *Computronix Python/Oracle Interface: cxOracle (http://www.cxtools.net/default.aspx?nav=cxorlb)*, cxtools.net.