Planning-Based Multimedia Adaptation Services Composition for Pervasive Computing

Girma Berhe[1], Lionel Brunie[1] and Jean-Marc Pierson[2]
[1] Laboratoire LIRIS UMR 5205, INSA de Lyon, [2] Laboratoire IRIT UMR 5505, Université Paul Sabatier, Toulouse 3

{girma.berhe1, lionel.brunie1}@liris.cnrs.fr, jean-marc.pierson@irit.fr

Abstract. Content adaptation is an attractive and effective solution to resolve the mismatch of resources and properties between the delivery context and the multimedia content in heterogeneous environments. The problem with multimedia content adaptation is that there is no a single complete software solution that can satisfy all types of required adaptation needs. In order to solve this problem adaptation tools can be developed as services for example using Web Services and make them accessible via standard web protocols. Since adaptation is a multi-step process, an adaptation need can be realized as composition of a number of adaptation services. However, it becomes difficult when there are several services to realize an adaptation need which leads to different composition possibilities. In this paper we present an algorithm called Multimedia Adaptation Graph Generator (MAGG) that composes distributed multimedia adaptation services. The algorithm is tested in a Distributed Content Adaptation Framework (DCAF) prototype and its experimental result is presented.

Keywords: Content adaptation, Context profile, transformation graph, Service Composition, Pervasive Computing.

1 Introduction

As users are beginning to rely more heavily on pervasive computing devices, there is a growing need for applications to bring multimedia information to the devices. However, due to limited device capabilities in terms of the display size, storage, processing power, and network access there are new challenges for designing the applications that allow these devices to effectively access, store and process multimedia information. Concurrent with the developments in pervasive computing, advances in storage, networking and authoring tools are driving the production of large amounts of rich multimedia content. The result is a growing mismatch between the available multimedia content and the capabilities of the client devices to access and process it. This mismatch impacts a number of multimedia applications.

Content adaptation is an effective and attractive solution to solve the mismatch problem, resource limitation and satisfy user's preferences in heterogeneous environments such as pervasive systems [8]. Content adaptation or multimedia content adaptation refers to the process of customizing (tailoring) and transcoding content to suit to the user's computing environment such as location, device capability, network bandwidth, user's preferences and usage context [14, 9].

Since content adaptation has become one of the research interests in pervasive computing, different approaches have been proposed: server-based [15, 17], client-based [16, 12] and proxy based [18, 19, 11]. While server-based adaptation degrades the performance of the server, and client-based adaptation is very difficult and sometimes impossible due to limited processing power of pervasive devices such as Smart phone and PDAs, most of existing adaptation systems chooses the proxy-based approach. To alleviate the overload problem of content adaptation processes, distributed approaches were proposed such as Ninja [7], MARCH [1] and DANAE [10].

However, these approaches do not provide a general adaptation solution. This leads to a service-based distributed adaptation where adaptation tools are developed externally as services and accessible through standard web protocol [2]. Major reasons for the need to externalize adaptation processes are (1) it is difficult to have a single complete software solution that can provide all the required multimedia content adaptation (transformation) tools by one system (2) using Web technologies such as Web Services, adaptation tools can be easily developed and make them accessible via standard web protocols (3) by using third party specialized adaptation tools, performance of the process can be enhanced at the same time good adaptation result can be achieved as specialized server provides better result than a general one (4) finally, content adaptation is a multi-step process hence an adaptation need can be realized as a composition of several adaptation services.

The rest of the paper is organized as follows: in Section 2, we briefly present some of the related works in content adaptation approaches. In Section 3, we present system architecture of Distributed Content Adaptation Framework (DCAF). The Multimedia Adaptation Graph Generator (MAGG) algorithm is presented in Section 4. MAGG implementation in DCAF Prototype is presented in Section 5. Section 6 presents performance evaluation of the algorithm. Finally, we present the conclusion in Section 7.

2 Related Work

The Content adaptation has been actively explored by many research groups. In general these works can be categorized into two main approaches: static and dynamic [13]. In static adaptation, contents in formats specific for each terminal or class of terminals are created and stored by the content provider. This approach has several advantages in terms of delivery performance, specificity to particular device, and full control of the content by the server; however it lacks consideration of dynamic variations of terminal's hardware or software components. Moreover, authoring and updating of the content is very costly and impractical and also requires large storage space [12]. To address such issues dynamic content adaptation is proposed. In dynamic content adaptation, the system gathers characteristics of the terminal, user and network, called delivery context or context profile, and adapts the content accordingly. There are ongoing standardization activities to develop an open framework of multimedia delivery and consumption. The work in progress by MPEG-21 is one of them. The aim of MPEG-21 is to enable the use of multimedia resources across a wide range of networks and devices [3]. One part of MPEG-21 is the Digital Item Adaptation. DIA defines description tools for usage environment and content format features for the purpose of transparent access to the multimedia content. These tools enable the adaptation engine to make better decision in order to provide best content to the user.

Although the previous works has shown that the use of the general adaptation approaches do result in better performance and user experience compared to situations when no adaptation is provided, each of these approaches exhibits performance shortcoming in certain environments [6]. Moreover, these approaches limit openness, flexibility, and interoperability of the data delivery system. In [2], we proposed a service-based architecture (DCAF). In this architecture adaptation tools are developed externally by third party or service providers. The idea is the service providers can develop the adaptation tools for example using Web Service technology and can be accessible by data delivery applications. As stated by [8, 20], external services enhance the performance of the adaptation process and accommodate a number of adaptation requirements which is not the case in previous approaches.

3 System Architecture

Fig.1 describes the DCAF architecture. For details on the architecture see [2]. Main components of the architecture and their function are summarized as follows:
Local Proxies (LPs): access points to information delivery systems. They are in charge of retrieving and processing context profile (user, terminal, network), decide the type and number of adaptation processes, discover appropriate adaptation services and plan execution of the services.
Context Profile Repository (CPR): stores user and device profile. Users can update and modify their profiles at any time. Dynamic profiles such as user location and networking condition are determined during request execution.
Adaptation Service Registry (ASR): it is like Universal Description, Discovery and Integration (UDDI) registry; it stores adaptation services profile (functional and non-functional) and allows for service look up.
Adaptation Services (ASs): to keep the framework open and extensible, we consider external adaptation tools (adaptation services). The ASs hosts the adaptation tools.
Content Proxies (CPs): provide access to content servers, formulate user request to source format, manage and provide content profile (meta-data).
Content Servers (CSs): they are standard data repositories such as web sites, databases, and media servers.

The resource mismatch comes from the fact that on one hand most of the devices have limited capabilities in terms of processing, storage, display etc. on the other hand the multimedia data require large resource. The format mismatch is resulted from the different software and hardware installation of each device and the existence of different data formats of the multimedia data. Content adaptation is an effective and attractive solution to solve the mismatch problem, resource limitation and satisfy user's preferences in heterogeneous environments such as pervasive systems [8]. Content adaptation or multimedia content adaptation refers to the process of customizing (tailoring) and transcoding content to suit to the user's computing environment such as location, device capability, network bandwidth, user's preferences and usage context [14, 9].

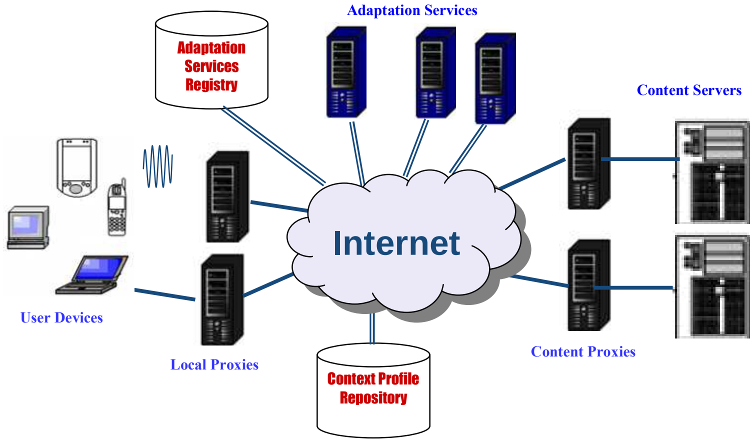

Fig. 1. Distributed Content Adaptation Framework (DCAF)

4 Multimedia Adaptation Graph Generator (MAGG)

In a distributed content adaptation like DCAF, the challenge is to find an appropriate service composition plan that provides the required adaptation need. Composition is defined as the process of putting together atomic/basic services to perform complex tasks. Given the fact that a composite adaptation task can be carried out in a number of steps (adaptation tools) and there could be different possible realization of the task makes service composition difficult. Moreover, finding an optimal adaptation path from the composition is also another problem. In order to solve these problems we have developed a composition algorithm. We call this algorithm a Multimedia Adaptation Graph Generator (MAGG).

4.1 Definitions and Notations Used in MAGG

Definition 1: Media object

A media object is a multimedia data which can be a text, an image, an audio or a video represented as

$M\ (m_1, m_2, \ldots, m_n)$

where $m_1, m_2, \ldots, m_n$ are media features or metadata

Definition 2: State

The state *S* of a media object *M* say *S* (*M*) is described by the metadata values

Example: for an image object the state holds the values for the format, the color, the height, the width, etc.

e.g. S(M) =(bmp, 24 bits, 245 pixels, 300 pixels).

Definition 3: Adaptation task

An adaptation task is an expression of the form t (a_1 a_2 ...a_n) where t is a transformation and a_1, a_2, ...a_n, are parameters

For example, ImageFormatConversion(imageIn, imageOut, oldFormat, newFormat), where

imageIn: image input file
imageOut: image output file
oldFormat: old file format
newFormat: new file format

Definition 4: Adaptation service

An adaptation service is a service described in terms of inputs, outputs, preconditions and effects. An adaptation service is represented as *s*= (*R I O Pre Eff Q)*

where
R: an atomic process that Realizes an adaptation task
I: input parameters of the process
O: output parameters of the process
Pre: preconditions of the process
Eff: effects of the process
Q: quality criteria of the service

Definition 5: Operator (plan operator)

A plan operator is an expression of the form *o*= (*h* (a_1, a_2, ...a_n, b_1, b_2, ...b_m), *Pre Eff Q*) where

h: an adaptation task realized by an adaptation service with input parameters a_1, a_2, ...a_n and output parameters b_1, b_2, ...b_m. h is called the head of identification of the operator.

Pre: represents the operator's preconditions.

Eff: represents the effect of executing the operator.

Q= {q_1, q_2,...,q_n}: represents quality attributes (e.g. cost, time, etc.)

Let *S* be a state, *t* be an adaptation task and *M* a media object. Suppose that there is an operator *o* with head h that realizes *t* such that *Pre* of *o* is satisfied in *S*. Then we say that *o* is applicable to *t* and the new state is given by:

$$S(M_o) = Executing(o, M, S)$$

Example: for the above given adaptation task, we can have an adaptation operator instance as follows:

Operator: ImageFormatConversionOperator (http://media-adaptation/imagefiles/image1, http:// media-adaptation/imagefiles/image2, mpeg, bmp)

Input: http://media-adaptation/imagefiles/image1

Output: http:// media-adaptation/imagefiles/image2

Precondition: hasFormat (http://media-adaptation/imagefiles/image1, mpeg)

Effect: hasFormat (http:// media-adaptation/imagefiles/image2, bmp)

Quality: (cost=30 units, time=100 ms)

Note: An adaptation task *t* can be realized by several adaptation operators hence *Q* will be used to compute an optimal plan or path.

Definition 6: Adaptation graph

An adaptation graph *G* (*V*, *E*) is a Directed Acyclic Graph (DAG) where

- *V* is the set of nodes that represent the adaptation operators
- *E* is the set of edges that represent the possible connections between the adaptation operators.

The start node **A** $\in$ *V* is a pseudo operator with effect (initial state) but no precondition

The end node **Z** $\in$ *V* is a pseudo operator with precondition (goal state) but no effect

Remark:

Let $o_i \in V$ and $o_j \in V$, a link or an edge e_{ij} exist from o_i to o_j if the following condition is satisfied:

$o_j.Pre \subseteq o_i.Eff$

Where

- $o_j.Pre$ denotes preconditions of o_j
- $o_i.Eff$ denotes effects of o_i

Definition 7: Adaptation planning problem

An adaptation planning problem is a four-tuple (S_A, S_Z, T, D) where S_A is the initial state of the media object, S_Z is the goal state of the media object, *T* is an adaptation task list and *D* is the adaptation operators. The result is a graph G= (V, E).

Definition 8: Adaptation path

An adaptation path is a path in the adaptation graph G that connects the start node to the end node. It is represented as a list of the form *p*= (**A** o_1 o_2 … o_n **Z**)

Where

A and **Z** are the start and the end nodes and o_i is an adaptation operator instance.

If p= (**A** o_1 o_2 … o_n **Z**) is a path and S is a state, then $S(p)$ is the state of the media object produced by executing o_1, o_2, …, o_n in the given order on the input media object.

4.1 Optimal Adaptation Path Search

Since an adaptation task can be achieved by more than one service[1] and each service has different QoS, choosing an appropriate service is an obvious requirement. Once the adaptation graph that consists of all possible compositions is generated, then the choice of the best alternative i.e. the optimal path in the graph is done based on user specified QoS criteria [2]. The QoS model contains two features: the execution and transmission time and cost of the service.

To find the optimal path one possible solution is to perform an exhaustive search that is, computing all possible adaptation paths and selecting the optimal path. Obviously, such an approach is of combinatorial complexity (in dependency of the number of adaptation operators n in the adaptation graph) and only feasible for a small number of n. The Dijkstra's algorithm [5] can be used to find the optimal path. We can express the running time of Dijkstra's algorithm on a graph with E edges and V nodes as a function of |E| and |V| using the Big-O notation.

The running time for the simplest implementation of the Dijkstra's algorithm is O ($|V|^2$). With a binary heap, the algorithm requires O ((|E|+|V|) log (|V|)) time, and the Fibonacci heap improves this to O (|E| + |V|log(|V|). The algorithm can be made more efficient in the special case when the directed graph is acyclic. This can be achieved by modifying the algorithm to include a step for sorting the graph topologically. Since the adaptation graph is a Directed Acyclic Graph (DAG), the modified algorithm was used. The algorithm with a topologically sorted graph has a running time of O (|E|+|V|) which of course has less running time compare to the other implementations.

5 MAGG Implementation in DCAF

In implementing the MAGG algorithm different class object have been defined. Due to space limitation, we present only some of the classes and their definitions. The partial class hierarchy of the MAGG algorithm is presented in Fig.2.

[1] The services are represented by operators in the graph

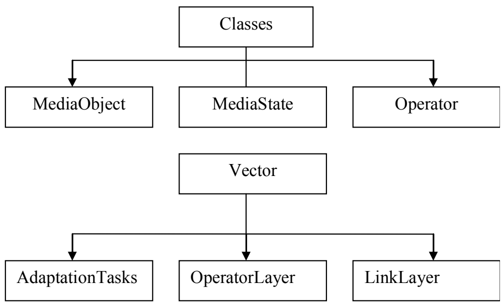

Fig.2. Class Hierarchy of MAGG

The MAGG algorithm was implemented in DCAF prototype. The prototype was developed in a medical application scenario. The prototype was built using Borland Jbuilder 9 personal. For the different databases developed in the prototype we used MySQL-4.0.12. The prototype provides pervasive data access to patient medical records. Fig.3 shows the conceptual data model used in the prototype.

Different graphical interfaces have been developed in order to facilitate the interaction of the system with the user. Fig.4 shows client profile entry interface.

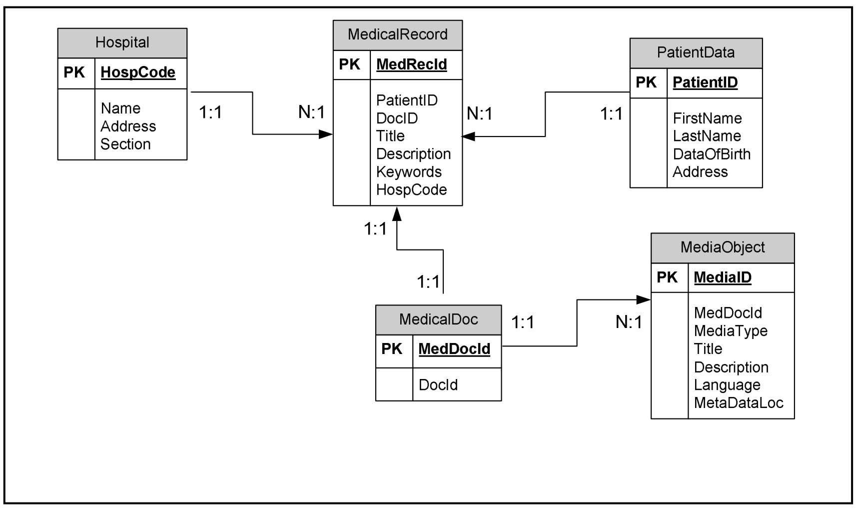

Fig.3. Conceptual data model

Different graphical interfaces have been developed in order to facilitate the interaction of the system with the user. Fig.4 shows client profile entry interface.

Fig.4. Client profile entry interface

The prototype displays the adapted data using SEFAGI [26] interfaces. The SEFAGI system provides an environment that permits the description of expected user interfaces using an abstract window description language, the description of target platforms using a platform description language and the automatic generation of user interface code based on the previous descriptions. Fig.5 shows an example of an adapted image (resized, format conversion and color reduction) on Smartphone.

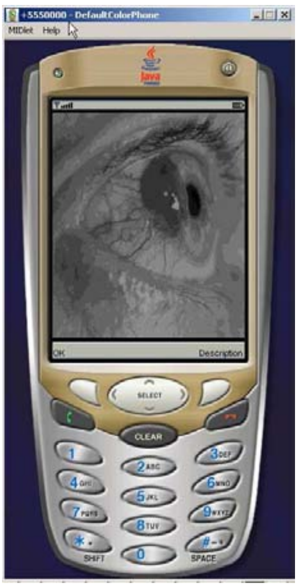

Fig.5. Adapted image on Smartphone

6 Performance evaluation

The experiment was to study the behaviour of the graph construction or generation algorithm with respect to the depth of the graph and the number of services per transformation. The graph construction time does not include the services execution time. The experiment was performed on a 1.9 GHZ Pentium 4 with 256 MB RAM running Microsoft Windows 2000. While the relationship between graph construction and the number of services per transformation is linear, compare to the graph depth, the construction time progress slowly with the number of service per transformation (see Fig.6). It was also observed that the progress both for the depth and the width (number of services per transformation) was almost constant with average increase of 40 ms for each depth and 10 ms for each 10 additional services. This implies that having several services per transformation does not affect much on the total construction time, while it provides the possibility to select the best service among the candidates. Most of the adaptation scenarios have 5 or 6 depth graph. Even for the maximum depth, for example, a total of 300 services (depth of 10 and width of 30) which is large enough to realize any possible type of adaptation need scenario; the construction time was only 335 ms which is really acceptable. Therefore, the overhead impact of construction time is less while on the other hand the algorithm is crucial in determining the optimal adaptation path required in delivering adapted data in pervasive data access.

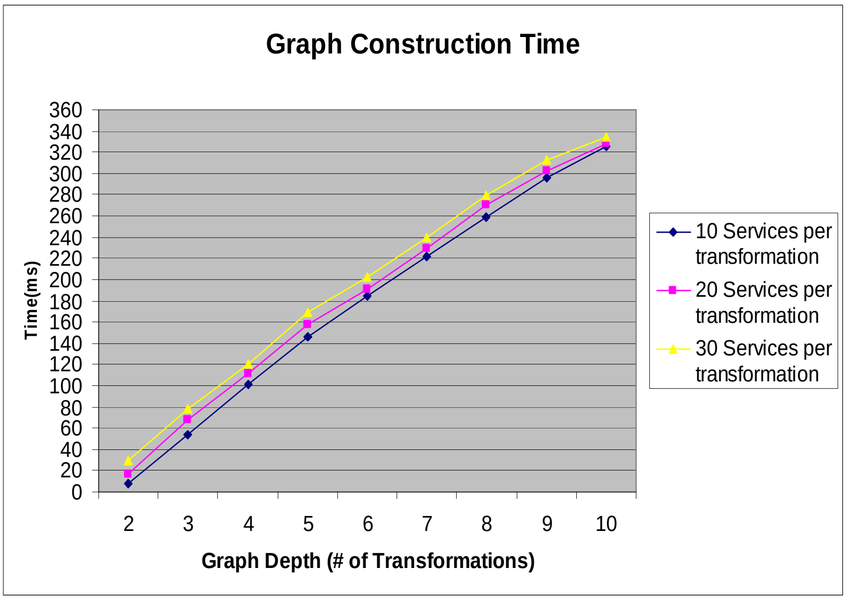

Fig.6. Graph Construction Time

7 Conclusion

In this paper we have presented an algorithm for multimedia adaptation graph generator. The algorithm uses planning techniques to compose adaptation services. The algorithm uses a two attributes QoS model to compute an optimal path. The QoS model consists of service cost and time. The algorithm was tested in a distributed content adaptation framework (DCAF) prototype. Construction time of the graph and computation of optimal path is insignificant with respect to the adaptation time. Hence, by incorporating the algorithm the system can find an optimal and effective service execution path.

References

1. S. Ardon, P. Gunningberg, B. LandFeldt et al. MARCH: A distributed content adaptation architecture. International Journal of Communication Systems, Special Issue: Wireless Access to the Global Internet: Mobile Radio Networks and Satellite Systems, 2003, Vol 16, No. 1, pp. 97-115.
2. G. Berhe, L. Brunie, J.M. Pierson. Content adaptation in distributed multimedia systems. Journal of Digital Information Management, Special Issue on Distributed Data Management, Volume 3, No 2, June 2005, pp.95-100.
3. I. Burnett, R. Van de Walle, K. Hill et al. MPEG-21: Goals and Achievements. IEEE Multimedia, October-December 2003, Vol.10, No.4, pp.60-70.
4. S. Chandra, C.S. Ellis, A. Vahdat. Differentiated Multimedia Web Services Using Quality Aware Transcoding. 19th Annual Joint Conference of the IEEE Computer and Communication Societies (INFOCOM 2000), Mar. 2000, Tel Aviv, Israel, pp. 961–969.
5. E. W. Dijkstra. A note on two problems in connection with graphs. Journal of NumerMath, 1959, Vol. 1, pp. 269-271.
6. X. Fu and V. Karamcheti. Performance Implications of Different Adaptation Mechanisms for Network Content Delivery. Accepted to the 23rd IEEE International Performance Computing and Communications Conference (IPCCC 2004), Apr, 2004, Phoenix, Arizona.
7. S. D. Gribble, M. Welsh, R. Behren et al. The Ninja architecture for robust Internet-scale systems and services. Journal of Computer Networks, March 2001, Vol. 35, No. 4, pp. 473-497.
8. A. Held, S. Buchholz, A. Schill. Modeling of Context Information for Pervasive Computing Applications. In Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI), July 2002, Orlando, FL, USA.
9. M. Hori, G. Kondoh, and K. Ono. Annotation-based web content transcoding. In Proceedings of the Ninth International World Wide Web Conference, May 15-19 2000, Netherlands, Amsterdam, pp. 197—211.
10. A. Hutter, P. Amon, G. Panis et al. Automatic Adaptation of Streaming Multimedia Content in A Dynamic and Distributed Environment. Proceedings of the IEEE Int’l Conference on Image Processing (ICIP 2005 2005), 11-14 Sept. 2005, Genova, Italy, pp. 716-719.
11. J.-G. Kim, Y. Wang, and S.-F. Chang. Content-adaptive utility-based video adaptation. In Proc. of IEEE Int'l Conference on Multimedia & Expo (ICME), July 6-9, 2003, Baltimore, MD, pp. 281-284.
12. Z. Lei and N.D. Georganas. Context-based Media Adaptation in Pervasive Computing. In Proc. of Canadian Conference on Electrical and Computer Engineering (CCECE), Toronto, May 2001.

13. W.-Y. Lum and F.C.M. Lau. A context-aware decision engine for content adaptation. IEEE Pervasive Computing, July-September 2002, Vol.1 No. 3, pp. 41-49.
14. W.-Y. Lum and F.C.M. Lau. On balancing between transcoding overhead and spatial consumption in content adaptation. Proc. of ACM Mobicom 2002, Sept. 2002, Atlanta, USA, pp. 239-250.
15. M. Margaritidis and G.C. Polyzos. Adaptation Techniques for Ubiquitous Internet Multimedia. Wireless Communications and Mobile Computing, 2001, John Wiley & Sons, Vol. 1, No. 2, pp. 141-164.
16. K. Marriott, B. Meyer, and L. Tardif. Fast and Efficient client-Side Adaptivity for SVG. ACM Press, 2002: In Proc. of WWW 2002, May 2002, Honolulu, Hawaii, USA, pp. 496–507.
17. R. Mohan, J.R. Smith and C.S. Li. Adapting Multimedia Internet Content for Universal Access. IEEE Trans. on Multimedia, 1999, Vol. 1, No. 1, pp.104–114.
18. A. singh, A. Trivedi, K. Ramamritham et al. PTC: Proxies that Transcode and Cache in Heterogeneous Web Client Environments. World Wide Web Journal, 2004, Vol. 7, No. 1, pp. 7-28.
19. S. Wee and J. Apostolopoulos. Secure scalable streaming and secure transcoding with JPEG-2000. IEEE International Conference on Image Processing (ICIP), 14-17 Sept. 2003, Barcelona, Spain, Vol.1, pp. 205-212. ISBN: 0-7803-7750-8.
20. R. Rahman, R.R. Menon and L. Rafalow. 2002. Enabling OPES to Use Web Service for Callout Service. Draft-rrahman-web-service-00.txt (work in progress), (May) (http://standards.nortelnetworks.com/opes/non-wg-doc/draft-rrahman-web-service-00.txt).
21. T. Chaari, F. Laforest. SEFAGI: Simple Environment For Adaptable Graphical Interfaces - Generating user interfaces for different kinds of terminals. In Proceedings of the Seventh International Conference on Enterprise Information Systems (ICEIS), 25-28 May 2005, Miami, USA, pp. 232-237. ISBN 972-8865-19-8.