CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Archipel - Université du Québec à Montréal

A Framework for Quality-Driven Delivery in Distributed Multimedia Systems

B. Kerhervé[1], K.K. Nguyen[2], O.Gerbé[3], B.Jaumard[4]

[1]Université du Québec à Montréal kerherve.brigitte@uqam.ca
[2]Université de Montréal nguyenkk@iro.umontreal.ca
[3]HEC Montréal olivier.gerbe@hec.ca
[4]Concordia University bjaumard@ciise.concordia.ca

Abstract

In this paper, we propose a framework for Quality-Driven Delivery (QDD) in distributed multimedia environments. Quality-driven delivery refers to the capacity of a system to deliver documents, or more generally objects, while considering the users expectations in terms of non-functional requirements. For this QDD framework, we propose a model-driven approach where we focus on QoS information modeling and transformation. QoS information models and meta-models are used during different QoS activities for mapping requirements to system constraints, for exchanging QoS information, for checking compatibility between QoS information and more generally for making QoS decisions. We also investigate which model transformation operators have to be implemented in order to support some QoS activities such as QoS mapping.

Keywords – Quality of Service (QoS), Quality-Driven Delivery (QDD), distributed multimedia system.

1. Introduction

Most applications today handle large amount of multimedia content, which is distributed across network. Multimedia applications can be classified based on four dimensions: the task they perform, the type of media they involve, the situation of operation (e.g., geographical dispersion of users) and the behavioral characteristics of users (e.g. user expectations, skills) [15]. The system architecture supporting such applications is heterogeneous, consisting of a large number of client machines, database servers, video servers or other specific servers, all interconnected through communication networks. These complex environments require the integration of system management mechanisms providing system scalability, application adaptation and quality of service (QoS) support [4].

QoS support was initially introduced in the field of telecommunication networks and multimedia systems [Aur98] and led to proposals for management strategies aimed at deciding whether and controlling how multimedia streams can be delivered to the user under some constraints. Generally these constraints are expressed in terms of system performance relative to media delivery and synchronization. The system then works to deliver the specified level of service and for that purpose transforms the users' requirements into various constraints mainly targeted to the transport system [9][12].

More recently it appears that QoS should be considered from a broader perspective and addressed in the context of internet applications and distributed and wireless computing [1]. We believe that new approaches and proposals should position the user at the center of the process allowing the expression of non-functional requirements concerning performance as before, but also concerning cost or more generally the quality of the delivered information. We can then talk about quality-driven delivery (QDD) where the quality requirements expressed by the user are taken into account in the different steps of the delivery [8]. In such a context, all the components of the distributed multimedia system have to contribute to offer QDD and each of them should include specific mechanisms to support QDD locally and to provide information for global, distributed decisions relative to QDD. Such decisions can take different forms such as resource allocation decisions, content-based adaptation rules or query-routing decisions.

In this paper, we propose a general framework for QDD in distributed multimedia environments. Although most of the discussion is conducted in the multimedia area, the application of this framework is not limited to multimedia delivery. It is intended to be applicable to most applications that require to deliver objects. For example, it can be applied to distributed database management systems, where the database server is expected to send (deliver) a query result (an object) to the user.

For this QDD framework, we propose a model-driven approach where we focus on QoS information modeling and transformation. QoS information models and meta-models are used during different QoS activities for mapping requirements to system constraints, for exchanging QoS information, for checking compatibility between QoS information and more generally for making QoS decisions. We also investigate which model transformation operators have to be implemented in order to support some QoS activities such as QoS mapping.

The rest of the paper is organized as follows. Section 2 presents the principles and activities for QDD and positions our approach in the context of video delivery services. Section 3 describes the different QoS parameters that compose our QoS information models. Section 4 presents the transformation performed on the information models. Section 5 discusses some experimental results of using QDD approach for making QoS decisions.

2. Quality –Driven Delivery Services

Quality-driven delivery refers to the capacity of a system to deliver documents, or more generally objects, while considering the users expectations in terms of non-functional requirements. To support QDD, mechanisms have to be integrated into the service or the application in order to adapt the quality of service to the user's requirements. Some approaches have been proposed for multimedia application adaptation, more specifically for adaptation to the technical infrastructure used for accessing multimedia objects [14]. Most of the existing approaches and mechanisms are more oriented towards resource allocation than user-perceived quality. We believe that it is time to consider maximizing the user-perceived quality as a main objective of application adaptation. Since the user-perceived quality can be influenced by several factors, we claim that all these factors should be integrated in a theoretical framework being the kernel of an adaptation engine. All the quality information involved in the adaptation process should be collected, integrated and managed in a homogeneous manner.

2.1 A Video Delivery Service

To illustrate the principles of QDD, we take the example of a video delivery service where users specify their quality preference according to three dimensions: the language of the audio sequence and the size and the frame rate of the video. The following figure presents a simplified view of this service. We identify three main modules: the quality information manager (QIM), the decision engine (DE) and the adaptation and delivery engine (ADE).

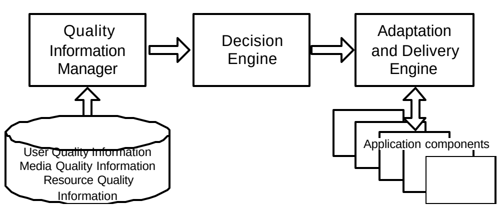

Figure 1: Components of a Video Delivery System

The QIM is in charge of collecting, storing, integrating and providing access to the quality information (QI) used by the decision engine. This information can be classified into three different categories: user QI describing the preferences and requirements of the user in terms of quality level; media QI describing the characteristics of the video sequences and finally resource QI describing the characteristics of the resources, such as user equipment, servers or network connections.

The decision engine, located at the center of the system, is in charge of making QoS decisions allowing the video delivery under the constraints specified by the users and/or concerning the available resources. Such decisions can be centralized [16] or distributed [13] and may lead to content adaptation, resource allocation or resource adaptation.

The ADE is responsible for executing the plan produced by the decision engine. The ADE interacts with the different components of the system (encoder, video server, network) to finally deliver the video sequence to the user.

As an example, we consider that the network bandwidth is decreasing, leading to a QoS violation. Different decisions can then be taken:

a) **Content optimization**: a possible content optimization technique is compressing the data content or changing the video codec.

For example, [2] proposes a technique replacing GIF images by JPEG images that may reduce transmission overload more than eight times. [19] presents a classification of content optimization techniques including: i) information abstraction for reducing bandwidth requirement, ii) data prioritization for providing different QoS levels, iii) modality transformation for transforming content adaptively to a particular device, iv) data transcoding for enabling universal access using pervasive computing device, and v) purpose classification for eliminating redundant information.

b) **Resource allocation**: a possible resource allocation decision is using RSVP to reserve more bandwidth on the path from server to client.

Actually, allocation and re-allocation are fundamental methods to address QoS violation problems. According to IETF (*Internet Engineering Task Force*), resource allocation is the first motivation in attempting to provide QoS over Internet by the Integrated Services model [5] served for setting up path and reserving resource in the network. In the context of multimedia network with multiple components, resource allocation is a multi-dimensional problem which should be addressed using the resource information profile, the application requirements and utility functions [17].

c) **Resource adaptation**: resource adaptation can lead to changing the network's path or the video server.

QoS adaptation is used to maintain, as long as possible, the service level agreement built at the negotiation phase and can be realized at the client or server side. During QoS violation, QoS adaptation is done transparently in real-time, in such a way that the system's transition takes place from one state to another one in order to provide the requested level of service.

It appears that resource allocation and adaptation can be combined to provide QoS in distributed systems [6]. Since an optimal decision is not easy to obtain, we believe that all the QoS dimensions have to be considered and that the decision engine should take the user requirements as the ultimate optimization objective.

3. Modeling Quality Information

In this section we describe quality information models for QDD services. The quality information models we propose are the kernel of quality information managers that have to be developed for QDD services. In this section, we describe the different elements of our models and Figure 2 presents them with a UML-like diagram.

3.1. Quality Informations

Quality information (QI) comes from different sources and can be heterogeneous. For example, QI associated to video objects can differ depending on the encoding format and the standard used to describe associated metadata. The monitoring tools used to collect QI about the service level of the system components can also produce heterogeneous information. In these two cases, we can see that there is a need for homogenization and integration of QI.

There is also a need for extension and adaptation of QI models as well as for tools allowing description, integration and translation of QI sets coming from different sources and represented using different formalisms or standards. This problem is similar to the problem of data migration or schema translation in the field of metadata management for data warehouses and web portals. For QDD, we are interested in a subset of metadata describing the quality of objects, data sources or resources. To solve the problem of data migration and schema translation, database researchers have recently proposed an approach based on model management [3]. They propose to address the problem of data migration and translation from a higher level of abstraction and to work on models rather than working on data. This approach would lead to the development of a generic infrastructure for managing models and to the introduction of model operations for integration and translation of data.

We believe that QI management for QDD is a good candidate for model management, because not only are we concerned by integration and translation of quality metadata but also because QDD is provided in a distributed and heterogeneous environment where monitoring tools are fully platform-dependant.

A QI model is built with the concept of quality dimension. Quality dimensions are used to describe objective or subjective characteristics relative to the quality level of the different actors of a delivery service or the quality level expected by the user. Subjective characteristics refer to the quality level perceived by the user while objective characteristics refer to a measurable quality level. An example of a quantitative dimension is *network_delay*. This quality dimension is objective and can be measured using monitoring tools for communication networks. An example of a subjective dimension can be *response_time* with the values: (unacceptable, bad, good, excellent). This dimension is qualitative since the possible values depend on the perception or the interpretation of the user.

A quality dimension takes its values in a definition domain. These values are used to build expressions associated to dimensions. Expressions can be the declaration of a value: for example *network_throughput = 1MB* or the declaration of a constraint such as *2ms <= response_time <= 5ms*.

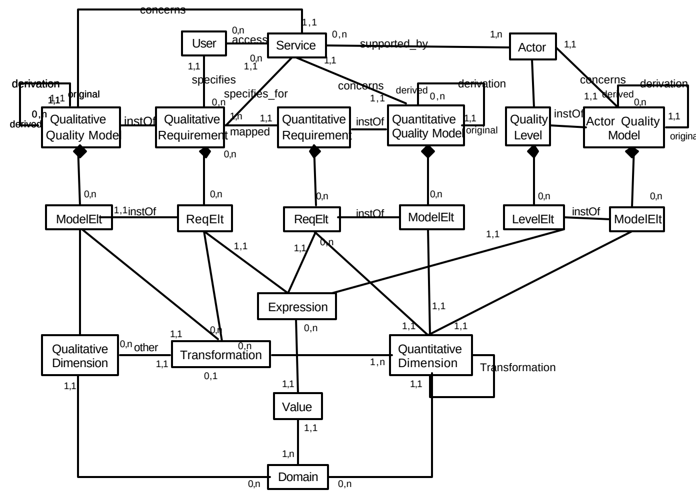

Figure 2: A UML diagram for QoS information model

3.2. Quality Information Models

The quality information, built with the concept of dimension, is modeled in quality information (QI) models. QI models describe the structure of quality information and allow the reuse, transformation and extension of existing models. QI models are composed of model elements, each of them describing a quality dimension. QI models can be User Quality Model, Actor Quality Model or Core Model [8].

The model elements of a User Quality Model describe the dimensions used to specify the expected quality level. We make a distinction between Qualitative Quality Model where the dimensions included in the model are qualitative dimensions, and Quantitative Quality Model where the dimensions are quantitative dimensions.

The model elements of an Actor Quality Model integrate the quality dimensions along which is described a quality level. We make a distinction between Media Quality Model built with the dimensions used to describe the quality level of an object to be delivered, and Resource Quality Model describing the quality level offered by a system component (communication network, database system, video server, user's device etc…).

Figure 3 presents the class hierarchy for Quality Information Model for the Video Delivery Service.

4. Model transformation

Model transformation allows the expression of relationships between the concepts of different quality information models. These relationships are defined on the quality dimensions and used to transform instances of a source model to instances of a target model.

The heterogeneity of QoS information between different layers and services can be dealt with different types of transformation through mapping operations. Mapping operations are essential for making QoS decision or information exchange. We identify two types of mapping, vertical and horizontal, used respectively for transforming information between layers and exchanging information between services of the same layer. From the point of view of QDD, these two types of mapping have to be implemented by the mapping between quality information models.

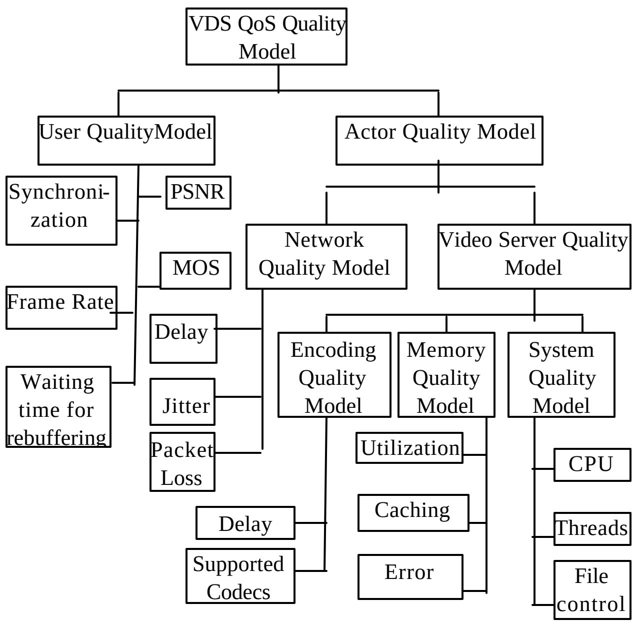

Figure 3: Quality information models of Video Delivery Service (VDS)

4.1. Vertical Mapping

Traditionally, four layers are used in QoS architectures [12]: the user layer, the service layer, the system layer, and the resource layer. In our general framework, User Quality Models correspond to the user and service layers. Actor Quality Models correspond to system and resource layers and are specialized to Network and Video Server Quality Models. The mapping between layers can be therefore modeled by different mappings between information models:

- from service layer to system layer: the projection of user requirements to system dimensions, performed by the translation of a User Quality Model in terms of an Actor Quality Model.

- from system layer to resource layer: by mapping Actor Quality Models to Network or Video Server Quality Models.

4.2. Horizontal Mapping

Mapping is also considered to respond to the needs of information exchange or service portability. While the vertical mapping is mostly used for layering architectures, the horizontal mapping is particularly useful for component-oriented architectures [10] where each system component acts as a black box with predefined input/output information. Services running on multiple platforms may also need horizontal mapping. In [11], the authors provided a sort of horizontal mapping to transform IP Diff Serv. specification into ATM QoS levels. In the context our QDD framework, the transformation between two platforms can be modeled by the mapping between instances of the corresponding quality information models.

Horizontal mapping can also be used to express that some decisions have equivalent impacts on the whole service. For example, increasing the video caching capacity or the network transmission rate can both lead to reduce the video rebuffering time. Therefore, the video rebuffering time can be expressed in terms of either network dimensions or server dimensions. From the user's point-of-view, these two resources (e.g. network and video server) are interchangeable and that could be described with the help of an horizontal mapping rule.

4.3. Mapping rules

Mapping activity is based on mapping rules, which explain how destination parameters are obtained from source parameters. Our mapping model classifies mapping rules into two categories:

- function-based rule: is basically a mathematical formula, often created using interpolation methods,
- table-based rule: is characterized by a lookup table, often defined by user or developer but as well built using experimental tests. A lookup table contains a limited number of entries by which a set of output values can be obtained from the input values.

4.3.1. Function-based mapping rules.

Function-based mapping is characterized by the accuracy and ease of implementation. The transport layer packet rate, inter-arrival time and end-to-end-delay can be expressed in terms of the delay, jitter and packet loss dimensions of the Network Quality Model using the function-based formulas provided by [9].

4.3.2. Table-based mapping rules.

Unfortunately, the number of function-based mapping rules is limited since building and validating a mapping function are usually a big challenge. Thus generally we do not have a sufficient number of function-based rules for realizing all the mappings in the system. This difficulty led us to use table-based rules as an alternative method. Table 1 shows an example of a table-based rule for mapping MOS to network bandwidth requirements [7]. While a function-based rule can be used in a wide range of applications or environments (i.e. packet rate calculation can be used in all the TCP/IP applications), a table-based rule is only valid within pre-defined working conditions, or within relatively stable states of a real system. This drawback is nevertheless limited since QoS is generally addressed in the context of a pre-defined application or system, such as the video delivery service previously presented.

MPEG-2 MOS*	Bandwidth (Mbps)
4.0	5.62
4.1	6.00
4.2	6.47
4.3	7.07
4.4	7.88
4.5	8.99
4.6	10.65
4.7	13.37
4.8	18.64
4.9	33.18

Table 1: Mapping from MOS to network bandwidth requirement

5. Experimentation and discussion

Based on our QDD framework, we conducted experiments for a video delivery service in the context of a QoS controllable environment, where a *perturbator* is implemented for handling some of the system QoS parameters. QoS decisions can be produced based on the client behavior which is observed in terms of the video frame rate, the image and audio quality as well as the re-buffering time. Our ultimate objective consists in building a set of rules covering all the possible mappings of user specification into dimensions of system components. These mapping rules are later used to make optimal decision concerning resource usage or adaptation strategies.

The behavior we observed at the client side is reported in Table 2. In this table, the first column represents the QoS parameter subject to change, the second column shows the treatment we applied to the parameter (increasing or decreasing), the third column describes the behavior observed at the client side, the fourth column explains the possible QoS decisions we can choose alternatively and finally the last column represents the QoS decisions in a optimal order (for our experimental environment).

Since traditional QoS approaches mainly focus on the communication network [20], when a QoS violation occurs, the preferred decisions are usually related to the network configuration (i.e. bandwidth re-allocation or server re-configuration). With our QDD approach, we take into account the QoS information of the overall system in order to make optimal decisions. For example:

- When delay is increasing, resulting in decreasing the video transmission rate, QoS decisions can be:
  - To allocate more bandwidth on the existing path. This costly decision is usually chosen in a provider-oriented QoS architecture;
  - To change the current transmitting server (or changing path). This decision requires further QoS information about the video servers in the system, but that may be worthy if the violation comes locally and uniquely from current server;
  - To change transmitting codec. This decision requires that QoS information about video codecs should be taken into account, but the overhead may be smallest.
- When jitter is increasing, resulting in video smooth layout , QoS decisions can be:
  - To increase temporary buffers of the transmitting servers or on-path network equipments. This decision is often costly;
  - To increase the receiving buffers at client side. This decision is quite simple but needs further QoS information about client side.
- When packet loss rate is increasing, resulting in poor video image/audio quality, QoS decisions can be:
  - To change transmission protocol (e.g. UDP to TCP since TCP is more reliable). Impacts of this decision on the client and server are important;
  - To change transmitting codecs (e.g. some codecs are more sensible to loss than others [18]). This decision is simpler, but requires QoS information about video codecs and selection algorithms should be implemented.
- When the caching time is increasing, resulting in video rebuffering, QoS decisions can be:
  - To allocate more bandwidth so that the transmission rate is increasing. This decision is costly;
  - To increase the buffer capacity on the client side. This decision is simple but needs a negotiation with user.
  - To change the video codec, that can lead to a smaller size of audio/image frame but also reduce video quality.

6. Conclusion

In this paper, we have introduced a general framework for Quality-Driven Delivery (QDD) in distributed multimedia environments. This framework follows a model-driven approach where we focus on QoS information modeling and transformation in order to take into account the contribution of all the system components for satisfying user's demands. We have presented the QoS information models and investigated different model operations to be implemented in order to support QoS activities. We also discussed about the experimentation of our QDD approach for video delivery service. In this discussion, we explained how optimal QoS decisions can be produced while taking into account the contribution of the different system components.

* MOS - *Mean Opinion Score*

Parameter	Treatment	Client behaviour	Possible QoS decisions	QoS delivery (reordering decisions)
Delay	↑	Transmission rate ↓	1.Allocate more bandwidth 2.Change video server 3.Change video codec	1.Change video codec 2. Change video server 3. Allocate more bandwidth
	↓	Transmission rate ↑	Not considered	
Jitter	↑	Video sequence smooth ↓	1.Increase server buffer 2.Increase client buffer	1. Increase client buffer 2.Increase server buffer
	↓	Video sequence smooth ↑	Not considered	
Packet loss	↑	Image/audio quality ↓	1.Change protocol UDP→TCP 2.Change codec	1.Change codec 2.Change UDP→TCP
	↓	Image/audio quality ↑	Not considered	
Caching	↑	Rebuffering time ↓	Not considered	
	↓	Rebuffering time ↑	1.Allocate more bandwidth 2.Increase client buffer 3.Change video codec	1. Increase client buffer 2. Change video codec 3. Allocate more bandwidth

Table 2: Changing network QoS parameters and making QoS decision (- : increasing, ¯ : decreasing)

7. References

[1] Abdelzaher, T., Shin, K., Bhatti, N., "User-Level QoS-Adaptive Resource Management in Server End-Systems", *IEEE Transactions on Computers*, Vol. 52, No. 5, May 2003.

[2] T. F. Abdelzaher, N. Bhatti, "Web Content Adaptation to Improve Server Overload Behavior", *Computer Networks*, Amsterdam, Netherlands, 1999.

[3] Bernstein, P., "Applying Model Management to Classical Meta Data Problems", *First Biennial Conference on Innovative Data Systems Research (CIDR 2003)*, USA, Jan. 2003, 209-220.

[4] Bochmann G., Kerhervé B., Lutfiyya H, Salem M, Ye H., "Introducing QoS to Electronic Commerce Applications", *The 2nd International Symposium on Electronic Commerce (ISEC)*, Hong-Kong, April, 2001.

[5] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource ReSerVation Protocol (RSVP) -Version 1 Functional Specification", *RFC 2205*, Sept. 1997.

[6] I. Foster, A. Roy, V. Sander, "A Quality of Service Architecture that Combines Resource Reservation and Application Adaptation", 8th International Workshop on Quality of Service, 2001.

[7] Fukuda, K., Wakamiya, N., Murata, M., Miyahara, H. "Effective Algorithms for Multicast Video Transport to Meet Various QoS Requirements", *IEICE Trans.Commun., vol. E81-B, no. 8, 1599-1607*, Aug. 1998.

[8] Gerbé, O., Kerhervé, B. Srinivasan, U. , "Model Operations for Quality-Driven Multimedia Delivery", *International Conference on Conceptual Structures, ICCS 2003*, Dresden, Germany, July 21-25, 2003.

[9] Huard J.-F., Lazar A. A, "On End-to-End QoS Mapping", *In Proceedings of the International Workshop on QoS*, May 1997, 303-314.

[10] Krishnamurthy, Y., Kachroo, V., Karr, D., Rodrigues, C., Loyall, J., Schantz, R., and Schmidt, D, "Integration of QoS-enabled distributed object computing middleware for developing next-generation distributed applications", *In Proceedings of the ACM SIGPLAN Workshop on Optimization of Middleware and Distributed Systems* ,Snowbird, UT, June 18, 2001.

[11] Lai, C., Tang, C., Chang, W., "Services mapping between Diffserv architecture and ATM", *Taiwan Academic Network Conference*, Taiwan, 2001

[12] Li, B., Kalter, W., Nahrstedt, K, "A Hierarchical Quality of Service Control Architecture for Configurable Multimedia Applications",*Jour. of High Speed Networks*, IOS Pres, 2001.

[13] S. Lima, P. Carvalho, and V. Freitas, "Distributed Admission Control for QoS and SLS Management", *Journal of Network and Systems Management - Special Issue on Distributed Management*, 12(3), September 2004.

[14] Margaritis, M., Polyzos, G., "Adaptation Techniques for Ubiquitous Internet Multimedia", *Wireless Communications and Mobile Computing Vol.: 1, 2*, 4/6 2001,. 141 - 163

[15] Miras, D., "Network QoS Needs of Advanced Internet Applications", *Internet2 - QoS Working Group*, 11/2002.

[16] K. Nahrstedt, D. Xu, D. Wichadakul, and B. Li, "QoS-Aware Middleware for Ubiquitous and Heterogeneous Environments ", *IEEE Comm. Magazine*, vol. 39, no. 11, pp. 2-10, 2001.

[17] Ragunathan Rajkumar, Chen Lee, John Lehoczky, Dan Siewiorek, "A Resource Allocation Model for QoS Management", *IEEE Real-Time Systems Symposium*, 1997.

[18] Robustelli, L., S. Loreto, A. Fresa, M. Longo, D. Spinelli, "Prototype of an Adaptive Voice Coder for IP Telephony", *International Conference on Software, Telecommunications and Computer Networks – SoftCom 2003*, October 7-10, 2003, Split Dubrovnik (Croatia), Ancona, Venice (Italy), pp. 859-863.

[19] Shaha, N., Dessai, A., Parashar, M., "Multimedia Content Adaptation for QoS Management over Heterogeneous Networks", *International Conference on Internet Computing*, Nevada, USA, Computer Science Research, Education, and Applications (CSREA) Pres, pp 642 – 648, 6/2001.

[20] D. Schmidt, D.Levine, and C. Cleeland, "Architectures and Patterns for High- performance, Real-time CORBA Object Request Brokers", *In Advances in Computers*, Marvin Zelkowitz, Ed., Academic Press, 1999.