JOURNAL OF OBJECT TECHNOLOGY

Online at http://www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©**JOT, 2005**

Vol. 4, No. 5, July - August 2005

A Framework for the Integration of Multimedia Data

Hyon Hee Kim, LG Electronics, Seoul, Korea
Seung Soo Park, Department of Computer Science and Engineering, Ewha Womans University, Seoul, Korea
Won Kim, SamSung Electronics, Seoul, Korea

Abstract

In this paper, we propose a unified framework, that we call UniMedia, for semantics-based integration and management of multimedia data. The framework is an adaptation and extension of the federated database technology that has been developed during the past three decades for integrating disparate data sources of alphanumeric data. In particular, we propose a multimedia data model, that we call XML/M, which encompasses diverse types of multimedia data and captures semantic relationships among them. We develop the concept of container objects which cluster relevant multimedia data and customize them according to users' preferences. Further, we augment the concept of container objects with a version management technique. To validate utility of the UniMedia framework, we have implemented a prototype multimedia news management system.

Keywords: XML-based multimedia data model, container mechanism for multimedia data, version management of multimedia data, multimedia application development framework

1 INTRODUCTION

With recent progress in computing technologies, multimedia data such as images, video clips, animations, graphics, and audio have proliferated over the past several years. Most Web-based applications have been developed including diverse types of multimedia data. Users have begun to expect that multimedia contents should be just as easily accessed as alphanumeric data. They want to see video clips related to text article they read, listen to music contained in a video clip they see, and find relevant photo images that appear in a movie or news video clips. To support such user needs, it is important to provide integrated access to diverse types of multimedia data stored in disparate data sources. However, many multimedia applications today deal with multimedia data from disparate sources separately.

Cite this column as follows:Hyon Hee Kim, Seung Soo Park and Won Kim: "A Framework for the Integration of Multimedia Data", in Journal of Object Technology, vol. 4, no. 5, July-August 2005, pp. 27-35 http://www.jot.fm/issues/issue_2005_7/column3

The aim of this paper is to propose an XML-based framework, that we call UniMedia, for semantic integration and management of multimedia data to support the development of a wide variety of multimedia applications. The UniMedia framework is similar to the Chamois framework [8] which has been developed at Ewha Womans University. While the Chamois knowledge engineering framework helps development of enterprise business intelligence applications such as customer-relationship management, electronic commerce, business analystics, and information personalization, the UniMedia framework supports multimedia applications such as news-on-demand applications, e-learning systems, and digital libraries.

The contribution of this paper is in the development (and validation) of the architecture of a federated database system that allows dynamic integration of disparate multimedia data sources, such as database systems, multimedia data servers, multimedia applications, asset management systems, etc. Our UniMedia framework represents an adaptation and extension of the federated database technology that has been established during the past three decades for integrating alphanumeric data from disparate data sources, such as database systems and file systems [7].

The framework is based on a multimedia data model, that we call, XML/M (M for multimedia). XML/M is composed of media objects, relationship objects, and container objects. Media objects describe diverse multimedia contents, and relationship objects maintain explicit relationships among the media objects. Container objects are containers of semantically related media objects. They are customized according to user needs and profiles, and then delivered to the users. Further, the container objects are augmented with version management capabilities to support efficient authoring tasks. We implemented a prototype multimedia news management system, that we call UniMediaNews, to validate utility of the UniMedia framework.

XML-based standards for multimedia data have been defined. MPEG-7 [9] is a standard for describing multimedia content. Semantic descriptions in content description tools are closely related to the XML/M data model. The main differences between the semantic description of MPEG-7 and that of our data model is that a basic object in MPEG-7 is a real-world object such as a person in images or a mountain in a video clip, while a base object in XML/M is a media object such as an image or a video clip. Therefore, while spatio-temporal relationships in MPEG-7 are about constituent objects in a single multimedia object, the semantic relationships in our model focus on relations among diverse multimedia objects. MPEG-21 [10] is expected to become an open standard framework for multimedia delivery and consumption. While multimedia data are integrated into a digital item in a digital item declaration statically in MPEG-21, the UniMedia framework integrates semantically relevant multimedia data dynamically using container objects.

The remainder of this paper is organized as follows: In Section 2, we give an overview of the UniMedia framework. In Sections 3 and 4, we discuss the XML/M data model and container management, two cornerstones of the UniMedia framework. In Section 5, we present the UniMediaNews system for validating the UniMedia framework. In Section 6, we provide concluding remarks.

2 OVERVIEW OF THE UNIMEDIA FRAMEWORK

In this Section, we first present the architecture of the UniMedia framework. Figure 1 illustrates the basic architecture of the UniMedia Framework. The framework is largely composed of three components: multimedia data sources and adapters, a container management component, and a metadata management component. The bottom of Figure 1 depicts heterogeneous multimedia data sources and adapters. Multimedia data sources can be database systems, multimedia data servers, multimedia applications, asset management systems, etc. An adaptor is the intermediary between UniMedia and particular data sources, which does the translation of a query and update statement to naive retrieval and update commands.

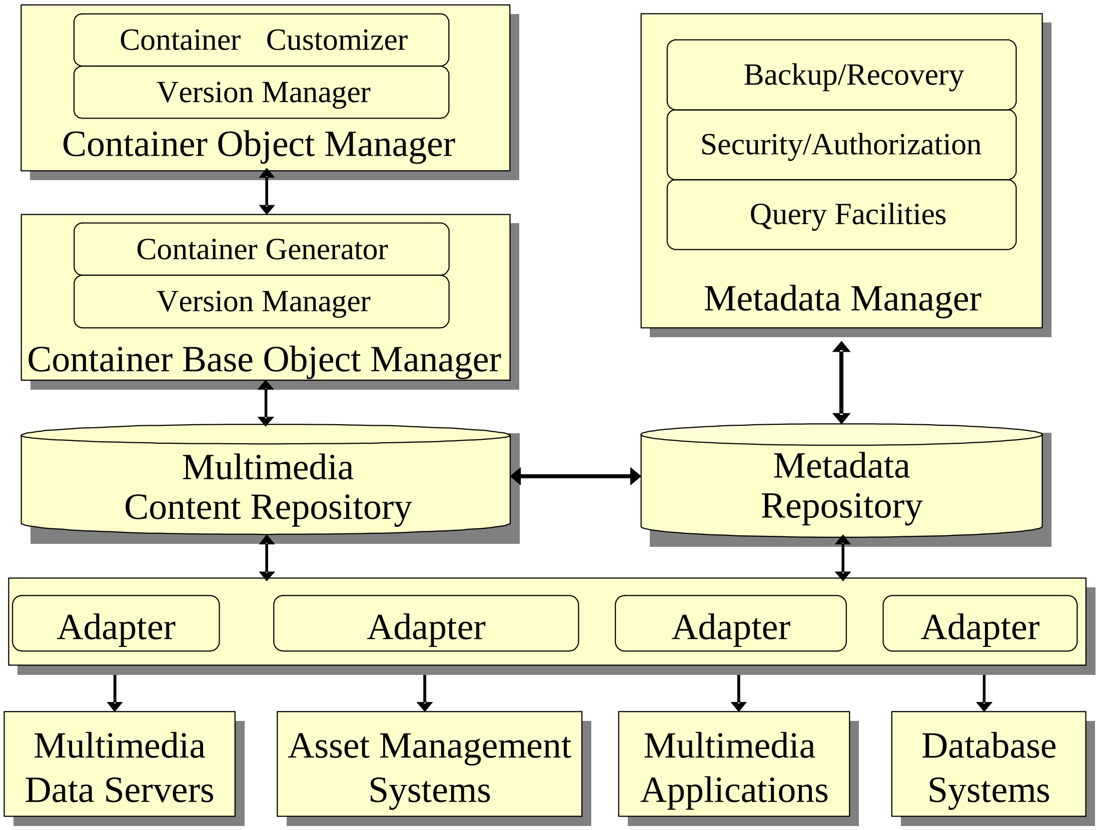


Figure 1. Architecture of the UniMedia Framework

The left side of Figure 1 shows the container management component which is composed of a container base object manager and a container object manager. The container base object manager is composed of a container generator and a version manager. The container generator creates container objects and stores them in the repository. The container objects are divided into single-media and multiple-media container objects. The single-media container objects are generated by restructuring and integrating homogeneous media objects, and are classified into image container objects, audio container objects, video container objects, and text container objects. On the other hand, the multiple-media container objects are generated by clustering different types of media objects based on relationships among media objects, and are classified into spatial container objects, temporal container objects, and semantic container objects.

Above the container base object manager, there is a container object manager, which is composed of a container customizer and a version manager. The container customizer customizes the container objects based on users' profiles. It transforms the customized container objects into a standard XML document or an HTML form using a pre-defined template. The version manager creates and manages versions of container objects through both the container generation and customization processes.

The right side of Figure 1 shows metadata management component. Metadata repository is the central storage area for UniMedia metadata, which provides UniMedia users with information about data sources. For management of the metadata, there are query facilities, security/authorization, and backup/recovery functionalities.

3 THE XML/M DATA MODEL

Due to its semistructured and self-describing characteristics, XML [1] has been widely used as a common data model for integrating heterogeneous data sources. Since XML is a markup language that allows user-defined tags, it is also useful to represent the content of various types of multimedia data. Therefore, we adopt XML as a common multimedia data model. To model multimedia data as objects using XML, we take two main features of the standard object-oriented data model [3]: *object identity* and *object nesting*. There are three types of objects in the multimedia content repository: media objects, relationship objects, and container objects.

Media Objects: A media object is the basic unit of multimedia data. For example, an image, an audio stream, and a video clip containing a meaningful scene are each modeled as a media object. Each media object is represented by an XML tree with a unique object identifier. The XML tree does not need a fixed schema, because it is self-describing. An image object might be described by the content of the image and metadata like color and shape, while a video object might be described by the content of the video clip like event, object and so on and metadata like duration and date.

Relationship Objects: A relationship object specifies the relationships among media objects. Managing relationships is one of the key features in many multimedia applications. The object-oriented data model captures the Is-A and Is-Part-of relationships between classes using a class hierarchy and a composition hierarchy, respectively [6]. However, multimedia data have more diverse relationships among media objects. We need to manage the relationship object as a first-class object, because we should be able to capture diverse relationships such as spatial relations, temporal relations, and non-spatio-temporal semantic relations and to represent 1-to-many and many-to-many relations as well as n-ary relations flexibly. While media objects focus on describing their contents, relationship objects are managed independently of media objects.

As with a media object, a relationship object is represented by a tree structure with a unique object identifier for the object. Whereas media objects do not have types for the tree structure, relationship objects do. A tree T is described by SpatialRelation, TemporalRelation, or SemanticRelation and each relation is described by a relation name and one or more participating media objects. A participating object (Pobject) is described by one or more audio objects (Aobject), image objects (Iobject), video objects (Vobject), or text objects (Tobject). Each participating object can be a media object or a container object that is a cluster of media objects. If a relationship object specifies a binary relation, it has two participating objects. According to the XML element order, the first object becomes the source object and the second object is the target object. If a relationship object specifies an n-ary relation, it can have several participating objects. In this case, we do not consider the element order.

Container Objects: In our previous paper [4], we designed a mechanism that can be applied to multimedia data. We have extended the mechanism to container objects. A container object is a cluster of semantically related media objects. There are two types of container objects: single-media and multiple-media container objects. Single-media container objects cluster semantically related homogeneous media objects, whereas multiple-media container objects integrate semantically related heterogeneous media objects. Like a relationship object, a single-media container object has a type for the tree structure. The type for the tree T is composed of two subtrees, and the subtrees describe the contents of the container object with a *semantics* element and the object identifiers of the participating media objects with *Pobject* element, respectively. Unlike those of relationship objects, the subtrees do not have fixed types.

Like single-media container objects, a multiple-media container object has a type for tree T. The type for a tree T is composed of two subtrees which describe a target object with *TargetObj* element and related objects with *RelatedObj* element, respectively. The *TargetObj* tree is composed of the object identifier of the target object and a *semantics* element, while *RelatedObj* is composed of diverse semantic relationship objects.

4 THE VERSION MANAGER

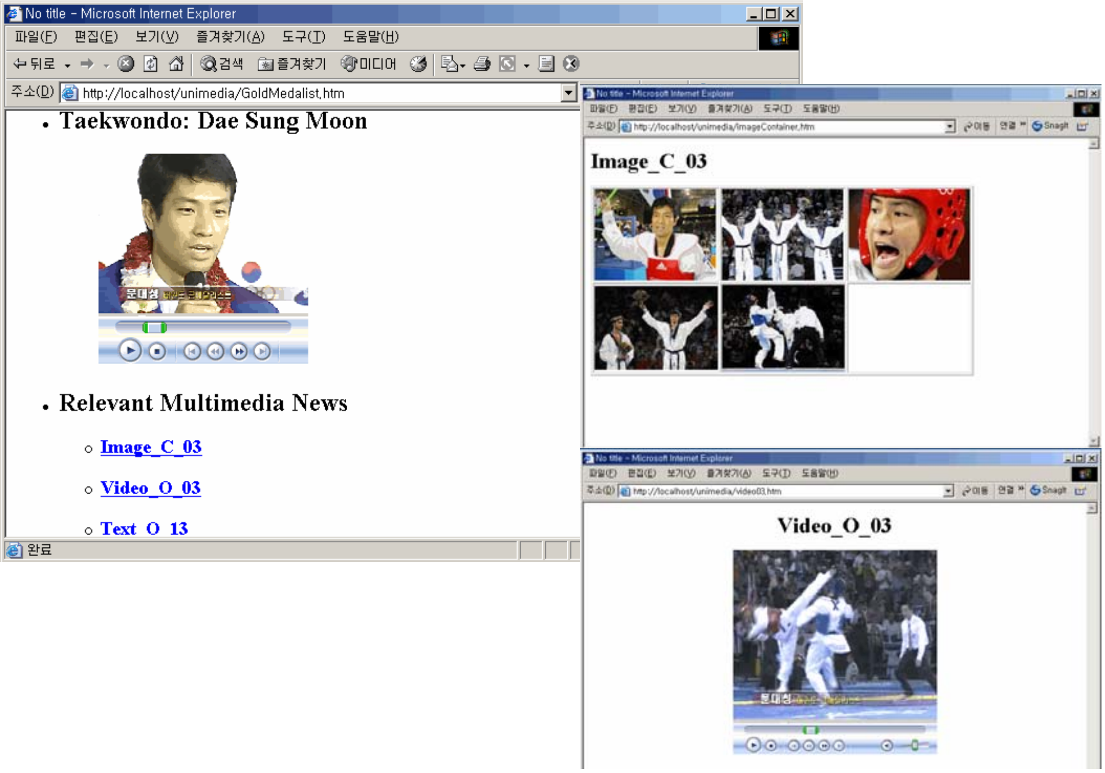
There is a general consensus that version control is one of the important functions in multimedia applications [2]. In our previous paper [5], we proposed a versioning scheme for efficient partial updates of XML documents. We adapt Chou and Kim's version model [5] to support a versioning scheme appropriate for UniMedia. The model does not include consideration of differential versioning, that is, keeping only the differences between the original and a version derived from it.

In order to allow partial update of a container object, a container object is transformed into two subtrees, that is, one to be changed (versionable subtree) and the other not to be changed (non-versionable subtree). Our approach is to share the non-versionable subtree and to apply versioning only to the versionable subtree. Version history is represented as *version trees*. UniMedia users can define an XSLT template for the transformation of a selected container object, and before versioning, execute the XSLT script. Versions of the versionable subtree are stored in the repository, and users can retrieve a version from the repository using XQuery just as ordinary objects are retrieved. If there are several versions that satisfy the query, the latest version is selected as a default. After a version is selected, users construct a full version of a container object by merging the non-versionable subtree and the selected version of the versionable subtree.

5 VALIDATION: A MULTIMEDIA NEWS MANAGEMENT SYSTEM

In order to validate the UniMedia framework, we have developed a multimedia news management system that integrates, updates, customizes, and delivers multimedia news items to customers. A news agency wants to build a multimedia news management system which integrates, manages, customizes, and delivers relevant multimedia news items. It has text articles, photo news, and video news in related database systems or video servers, separately. They want to integrate relevant multimedia news about a gold medalist at the Athens 2004 Olympic Games from the separate data sources, and to deliver the integrated multimedia news to different types of users such as newspapers, portal sites or groups of individual users.

Figure 2 shows an example customized container object. The video clip played in Figure 2 is a target object, a video clip containing an interview scene for an Olympic gold medalist, “Dae Sung Moon”, and the target object has three related objects, *Image_C_03*, *Video_O_03*, and *Text_O_13*. *Image_C_03* consists of 5 photos of the athlete, and *Video_O_03* is a video clip containing the athlete’s final game scene in the Olympics. *Text_O_13* is a text article about the athlete’s interview.


No title - Microsoft Internet Explorer
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
뒤로 검색 즐겨찾기 미디어
주소(D) http://localhost/unimedia/GoldMedalist.htm
http://localhost/unimedia/ImageContainer.htm
http://localhost/unimedia/video03.htm
완료

- Taekwondo: Dae Sung Moon
- Relevant Multimedia News
  - Image_C_03
  - Video_O_03
  - Text_O_13

Image_C_03

Video_O_03

Figure 6. An Example of a customized container object

The system is implemented on top of a commercial native XML server, Tamino version 4.1 in Windows 2000 professional environment. To support versioning of container objects, WebDAV server is integrated with a database system. User interfaces are implemented using JavaServer Pages (JSP), and the entire UniMedia framework has been implemented with 3,000 lines of JSP. Using the UniMedia framework, a developer with JSP programming skills can implement this example in three hours.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we presented an XML-based framework for semantics-based integration and management of multimedia data as an adaptation and extension of the federated database technology. The framework is based on the XML/M data model composed of media objects, relationship objects, and container objects. The data model is designed to cluster relevant multimedia data and to customize them according to users' preferences. Further, the concept of container objects is augmented with a version management model.

While most existing multimedia modeling techniques focus on modeling specific types of multimedia data, our data model unifies different types of multimedia data. Due to the flexibility and self-describing features of our data model, multimedia contents are easily described, and relationship objects capture diverse relationships among different types of media objects explicitly. Container objects integrate semantically relevant media objects, and provide users with rich semantic information about the underlying multimedia contents. To validate utility if our framework, we implemented a multimedia news management system on top of our framework. Our validation system, UniMediaNews, showed that the UniMedia framework enables significant productivity advantages in developing the news management application. The productivity advantages result from three facilities of UniMedia: personalization, semantic support, and versioning.

We are currently enhancing the multimedia semantics with ontology technologies by reasoning about implied relationships among the physical objects appearing in the media objects. Work on XML warehouses or XML-based mediators is still not mature, with many problems remaining to be solved [3]. In the context of UniMedia, performance issues such as indexing techniques and query optimization techniques should be further studied.

ACKNOWLEDGEMENTS

This work was supported in part by the Brain Korea 21 Project of Korean Ministry of Education and done while the first author was a visiting student at the Univervisy of Stuttgart, Stuttgart, Germany. She thanks Prof. Bernhard Mitschang for facility support and useful comments.

REFERENCES

[1] T. Bray, J. Paoli and C. M. Sperberg-McQueen, Extensible Markup Language, http://www.w3c.org/TR/XML

[2] H. Chou and W. Kim, A Unifying Framework for Version Control in a CAD Environment, In *Proceedings of the VLDB Conference*, pp. 336-344, 1986.

[3] H. Garcia-Molina et al. The TSIMMIS approach to mediation: data models and languages, *Journal of Intelligent Information Systems*, pp. 117-132, Vol. 8, 1997.

[4] H. H. Kim and S. S. Park, Mediaviews: A Layered View Mechanism for Integrating Multimedia Data, In *Proceedings of 9th International Conference on Object-Oriented Information Systems, LNCS* 2817, pp. 250-261, 2003.

[5] H. H. Kim and S. S. Park, A Semantics-based Versioning Scheme for Multimedia Data, In *Proceedings of DASFAA Conference, LNCS 2973*, pp. 277-288, 2004.

[6] W. Kim, Object-Oriented Databases: Definition and Research Directions, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 2, No. 3, pp. 327-341, 1990.

[7] W. Kim, Modern Database Systems: The Object Model, Interoperability, and Beyond, *ACM Press and Addison Wesley*, 1995.

[8] W. Kim and et al., The Chamois Component-Based Knowledge Engineering Framework, *IEEE Computer*, Vol. 35, No. 5, pp. 46-54, 2002.

[9] MPEG-7, http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm

[10] MPEG-21, http://www.chiariglione.org/mpeg/standards/mpeg-21/

About the authors

Won Kim is Senior Advisor at SamSung Electronics, Korea. He is Editor-in-Chief of ACM Transactions on Internet Technology (http://www.acm.org/toit), and Chair of ACM Special Interest Group on Knowledge Discovery and Data Mining (http://www.acm.org/sigkdd). He is the recipient of the ACM 2001 Distinguished Service Award. He can be reached at wonkim@austin.rr.com

Seung Soo Park is professor of computer science and engineering. His areas of interest include artificial intelligence, data mining and bioinformatics. He received a Ph.D. degree in computer science from the University of Texas at Austin, USA.

Hyon Hee Kim is a senior research engineer in the digital media laboratory at LG Electronics, Seoul, Korea. Her research interests include digital multimedia broadcasting, audio/video codec technologies, and XML. She received a Ph. D degree in computer science and engineering from Ewha Women's University, Seoul, Korea.