OBJECT-ORIENTED HARMONIZATION OF MULTIMEDIA XML APPLICATIONS

Yih-Feng Chen, May-Chen Kuo, Xiaoming Sun and C.-C. Jay Kuo

Integrated Media Systems Center and Department of Electrical Engineering
University of Southern California, Los Angeles, CA 90089
E-mails: yihfengc@usc.edu, maychenk@usc.edu, sunx@pollux.usc.edu, cckuo@si.pi.usc.edu

ABSTRACT

The problem of creating powerful XML applications by reutilizing and integrating functionalities from existing XML applications, called XML harmonization, is studied in this research. An object-oriented (OO) data model is first proposed. Then, the corresponding harmonization methodology for XML applications is presented. That is, we develop the Harmonized eXtensible Markup Language (HXML) according to the proposed OO data model. HXML provides an uniform interface for users to access functionalities and objects of various XML applications being harmonized. An example that harmonizes two multimedia XML applications, i.e. Synchronized Multimedia Integration Language (SMIL) and eXtensible 3D (X3D), using the OO technique has been implemented and compared with the eXtensible MPEG-4 Textual format (XMT). Our methodology provides better extendibility than any previous work in this emerging research field.

1. INTRODUCTION

XML has been used extensively in various domains such as multimedia applications and databases because of its flexibility and interoperability. Since building XML structures is easy, the number of XML applications grows rapidly and leads to many redundant specifications. While some XML applications have common functionalities, others may have quite different ones. For instance, both the SMIL and the X3D have sub-schemas to describe the meta information yet they also have their individual functionalities to represent multimedia files and create X3D objects. Consequently, there is a demand on creating powerful XML applications by reutilizing and integrating functionalities from the existing XML applications. This is called the XML harmonization problem.

Most previous work on XML harmonization has been done in the database field, where the focus has been on the storage of XML instances in legacy databases and the integration of several databases that have similar schemas. The amount of harmonization work on multimedia applications has been relatively less, e.g. [1, 2], where the focus was on harmonizing metadata whose structures are quite regular. New schemas were defined in [1, 2] for metadata by analyzing the participating XML applications. Their methods are application-specific and cannot be generalized to other XML applications easily. The XMT is another harmonization effort in the multimedia domain. XMT simply adopts partial schemas from SMIL and X3D by renaming their element tags [3]. Generally speaking, previous work on harmonization is simple and heuristic.

In this work, we present a systematic approach to XML harmonization with the following contributions. First, the XML harmonization concept is formalized mathematically. Second, an OO harmonization methodology is proposed for multimedia XML applications. This methodology is chosen since its concepts of classes, aggregation, and inheritance can be adequately utilized to modulize functionalities and objects. Finally, we provide a detailed treatment on the harmonization data model, the design of HXML, and harmonization algorithms with a concrete illustrative example so that the general principle can be more easily conveyed.

2. FORMALIZATION OF XML HARMONIZATION

By “harmonization of XML applications”, we mean the analysis, transformation, and integration of multiple XML schemas. It does not only merge different schemas but also makes heterogeneous XML properties interact with each other so as to provide a uniformly consistent interface to high-level applications. Moreover, the harmonized result can be customized according to different sizes of user requirements. The harmonization procedure can be viewed as applying a sequence of harmonization operators to a set of XML applications. This concept can be mathematically formalized as follows.

Let $\mathcal{X} = \{X_i | i \in \mathcal{N}\}$ be a set of XML applications, where $X_i = \{x_{ij} | 1 \leq j \leq n_i, i \in \mathcal{N}\} = \bigcup_{i=1}^{m} S_i | m_i \leq n_i$ represents the schema of XML application $i$, and where

This research was funded by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, Cooperative Agreement No. EEC-9529152.
\( x_{ij} \) denotes the semantic elements of XML schema \( i \) and \( S_{it} \) denotes the different semantic sets of \( X_i \). Each \( x_{ij} \) belongs to one semantic set. Depending on the application domain, the semantic elements can be functionalities, data structures, etc.

We use \( X_H = \{ x_{ij} | 1 \leq j \leq n_H, n_H \in \mathbb{N} \} \subset \mathcal{X} \) to denote the harmonized XML. The XML harmonization process corresponds to a function \( H : \mathcal{X}_i \rightarrow X_H, \mathcal{X}_i \subset \mathcal{X} \). Specifically, given \( X_i \) and \( H = \{ h_1 \circ \ldots \circ h_p | m \in \mathbb{N} \} \), the XML harmonization system can be described by

\[
X_H = H(X_1, \ldots, X_N) = h_p \circ \ldots \circ h_1(X_1, \ldots, X_N)
\]

where \( h_j(X_i) = \{ h_{jk} | h_{jk} : S_{ik} \rightarrow S'_{ik}, S_{ik} \subset X_i, S'_{ik} \subset X'_i \subset \mathcal{X}, 1 \leq k \leq m_i \} \) is the harmonization operator that specifies the mapping between \( X_i \) and \( X'_i \). The subscripts \( j \) and \( k \) of \( h_{jk} \) indicate the \( j^{th} \) harmonizing step and the \( k^{th} \) semantic set, respectively.

### 3. OO-BASED LOGICAL MODEL FOR XML HARMONIZATION

#### 3.1. Object-Oriented Methodology

One solution to XML harmonization is to map XML schemas and instances into various existing language domains and/or data structures, and to apply harmonization operations in these domains accordingly [4]. Due to the potential of extensibility and the support of consistency and reutilization, the OO method has been widely used in system analysis and design, and is adopted in this work.

Key concepts and terminologies of the OO method are briefly reviewed below. A **class** is the prototype for a group of objects with a similar structure and behavior, which can be defined by three typical ways, i.e., encapsulation, inheritance, and aggregation. The structure of a class is determined by the **member variables** representing the state of an object of that class. The behavior of a class is given by a set of **methods** associated with the class. **Encapsulation** is the modelling of a class byencasing in the member variables and the methods such that the unnecessary information is hidden from outside access. **Inheritance** is the mechanism to derive new classes from existing classes by extending their definitions, where a derived class (or “subclass”) is a child of the “base class” (or “superclass”). **Aggregation** is a composition technique for building a new object from one or more existing objects that support some or all of the new object’s required interfaces.

The philosophy of the OO methodology is to build up an extendible class hierarchy based on semantic elements stated above to support interoperability and reutilization so that various sizes of application requirements can be accommodated.

#### 3.2. Formalization of OO-based Harmonization

Based on the operational differences, harmonization operators \( h_i \in H \) can be classified into two categories: **integration** and **restructuring**, where integration combines functionalities and objects from different XML applications and restructuring reorganizes the resultant XML. Fig. 1 (a) illustrates the structures of some existing multimedia XML applications, where the inner modules can be viewed as the children of the outer modules. The general information, spatial information and special effects of multimedia files do not locate in the same subtrees. Consequently, the retrieval of the related information of a given multimedia object may demand a considerable amount of efforts as highlighted by arrows in Fig 1 (a), which will in turn degrade the execution performance. To solve this problem, we propose an OO harmonization data model to harmonize multimedia XML applications *i.e.*, SMIL, X3D, etc. To be more specific, the information related to multimedia objects are organized in an OO fashion to make each multimedia object self-contained as illustrated in Fig. 1 (b).

![Comparison of two XML structures.](image)

**Fig. 1.** Comparison of two XML structures.

The harmonization of multimedia XML applications can be formalized as follows. Let \( X_i = \{ F_i, O_i \} \) represent the schema of multimedia XML applications, where \( F_i = \{ f_{i1}, \ldots, f_{im_i}, f_{c_1}, \ldots, f_{c_p} \} \) is the set of functionalities of \( X_i \) and \( O_i = \{ o_{i1}, \ldots, o_{im_i}, o_{c_1}, \ldots, o_{c_q} \} \) is the set of the objects of \( X_i \). It means that \( F_i \cup O_i = \{ x_{i1}, \ldots, x_{im_i} \} \) and \( F_i \cap O_i = \phi \). We use \( \{ f_{c_1}, \ldots, f_{c_p} \} \) and \( \{ o_{c_1}, \ldots, o_{c_q} \} \) to represent the common parts of functionalities and objects, respectively. Then, the harmonization process can be written as

\[
X_H = H(X_1, \ldots, X_N) = h_2(h_1(X_1, \ldots, X_N)) = h_3(h_2(h_1(X_1), \ldots, h_1(X_N))) = \{ \bigcup_{j=1}^{N} h_{1j}(F_j), \bigcup_{j=1}^{N} h_{12}(O_j) \}
\]

With respect to the general formula in Eq. (1), each XML application has been divided into two semantic sets, *i.e.* objects and functionalities, under the OO framework. There are two harmonization operations \( h_1 \) and \( h_2 \) in Eq.
(2), which represent the restructuring and integration, respectively.

3.3. OO-based Harmonization Data Model

We distinguish the properties that are common to all multimedia types from those that are specialized to each type. Four classes are recognized to be common to all multimedia objects. They are: Descriptor, Functionality, Spatial Information, and Temporal Information as shown in Fig. 2. Each class aggregates related semantic elements from the harmonized XMLs. For instance, “Spatial information” class aggregates “<region>” from SMIL and “<route>” from X3D. Then, Multimedia Class, a generalization of multimedia objects, aggregates these four common classes.

With aggregation of the four common classes, the Multimedia Class becomes self-contained. That is, all information related to a multimedia object is accessible inside the class. Six multimedia subclasses (3D, Audio, Graphics, Image, Video, and Text) are derived from the Multimedia Class, which has Descriptor, Functionality, Spatial Information, and Temporal Information classes as its members. These children classes also possess some special members that characterize themselves. Finally, the harmonization class aggregates an arbitrary number of Meta, HFunct, and some specific multimedia classes, where Meta and HFunct are globally accessible classes. Meta defines abstract descriptions for multimedia objects and HFunct defines globally accessible methods to operate on the HXML objects.

The Unified Modeling Language (UML) is used to represent the framework of the proposed OO harmonization approach. The harmonization architecture as shown in Fig. 3. Each class aggregates related semantic elements from the harmonized XMLs. For instance, “Spatial information” class aggregates “<region>” from SMIL and “<route>” from X3D. Then, Multimedia Class, a generalization of multimedia objects, aggregates these four common classes.

With aggregation of the four common classes, the Multimedia Class becomes self-contained. That is, all information related to a multimedia object is accessible inside the class. Six multimedia subclasses (3D, Audio, Graphics, Image, Video, and Text) are derived from the Multimedia Class, which has Descriptor, Functionality, Spatial Information, and Temporal Information classes as its members. These children classes also possess some special members that characterize themselves. Finally, the harmonization class aggregates an arbitrary number of Meta, HFunct, and some specific multimedia classes, where Meta and HFunct are globally accessible classes. Meta defines abstract descriptions for multimedia objects and HFunct defines globally accessible methods to operate on the HXML objects.

The Unified Modeling Language (UML) is used to represent the framework of the proposed OO harmonization approach. The harmonization architecture as shown in Fig. 3. Each class aggregates related semantic elements from the harmonized XMLs. For instance, “Spatial information” class aggregates “<region>” from SMIL and “<route>” from X3D. Then, Multimedia Class, a generalization of multimedia objects, aggregates these four common classes.

With aggregation of the four common classes, the Multimedia Class becomes self-contained. That is, all information related to a multimedia object is accessible inside the class. Six multimedia subclasses (3D, Audio, Graphics, Image, Video, and Text) are derived from the Multimedia Class, which has Descriptor, Functionality, Spatial Information, and Temporal Information classes as its members. These children classes also possess some special members that characterize themselves. Finally, the harmonization class aggregates an arbitrary number of Meta, HFunct, and some specific multimedia classes, where Meta and HFunct are globally accessible classes. Meta defines abstract descriptions for multimedia objects and HFunct defines globally accessible methods to operate on the HXML objects.

The Unified Modeling Language (UML) is used to represent the framework of the proposed OO harmonization approach. The harmonization architecture as shown in Fig. 3. Each class aggregates related semantic elements from the harmonized XMLs. For instance, “Spatial information” class aggregates “<region>” from SMIL and “<route>” from X3D. Then, Multimedia Class, a generalization of multimedia objects, aggregates these four common classes.

With aggregation of the four common classes, the Multimedia Class becomes self-contained. That is, all information related to a multimedia object is accessible inside the class. Six multimedia subclasses (3D, Audio, Graphics, Image, Video, and Text) are derived from the Multimedia Class, which has Descriptor, Functionality, Spatial Information, and Temporal Information classes as its members. These children classes also possess some special members that characterize themselves. Finally, the harmonization class aggregates an arbitrary number of Meta, HFunct, and some specific multimedia classes, where Meta and HFunct are globally accessible classes. Meta defines abstract descriptions for multimedia objects and HFunct defines globally accessible methods to operate on the HXML objects.

The Unified Modeling Language (UML) is used to represent the framework of the proposed OO harmonization approach. The harmonization architecture as shown in Fig. 3. Each class aggregates related semantic elements from the harmonized XMLs. For instance, “Spatial information” class aggregates “<region>” from SMIL and “<route>” from X3D. Then, Multimedia Class, a generalization of multimedia objects, aggregates these four common classes.

With aggregation of the four common classes, the Multimedia Class becomes self-contained. That is, all information related to a multimedia object is accessible inside the class. Six multimedia subclasses (3D, Audio, Graphics, Image, Video, and Text) are derived from the Multimedia Class, which has Descriptor, Functionality, Spatial Information, and Temporal Information classes as its members. These children classes also possess some special members that characterize themselves. Finally, the harmonization class aggregates an arbitrary number of Meta, HFunct, and some specific multimedia classes, where Meta and HFunct are globally accessible classes. Meta defines abstract descriptions for multimedia objects and HFunct defines globally accessible methods to operate on the HXML objects.

The Unified Modeling Language (UML) is used to represent the framework of the proposed OO harmonization approach. The harmonization architecture as shown in Fig. 3. Each class aggregates related semantic elements from the harmonized XMLs. For instance, “Spatial information” class aggregates “<region>” from SMIL and “<route>” from X3D. Then, Multimedia Class, a generalization of multimedia objects, aggregates these four common classes.

With aggregation of the four common classes, the Multimedia Class becomes self-contained. That is, all information related to a multimedia object is accessible inside the class. Six multimedia subclasses (3D, Audio, Graphics, Image, Video, and Text) are derived from the Multimedia Class, which has Descriptor, Functionality, Spatial Information, and Temporal Information classes as its members. These children classes also possess some special members that characterize themselves. Finally, the harmonization class aggregates an arbitrary number of Meta, HFunct, and some specific multimedia classes, where Meta and HFunct are globally accessible classes. Meta defines abstract descriptions for multimedia objects and HFunct defines globally accessible methods to operate on the HXML objects.

3.4. HXML

The Harmonization eXtensible Markup Language (HXML) is an XML expression of the OO-based harmonization data model (see Fig. 2), where classes are represented by elements and inheritance and aggregation relationships are mapped into parent-child relations. Two basic principles are applied in generating HXML from the OO data model as given below.

- If A inherits B, B is the sub-element of A.
- If A aggregates \( (B_1, \ldots, B_n) \), \( (B_1, \ldots, B_n) \) are the sub-elements of A.

We use the partial HXML DTD as shown in Table 1 to illustrate the process. The element “Harmonization” has three groups of sub-elements corresponding to the aggregation of Meta, HFunct, and Specialized multimedia objects. The entity “HFunctRoot” is the root of the subtree representing the expendable HFunct class members. The 3D class inherits the multimedia class so that we can specify element “Hobj3D”, which has the child “Common” that represents the multimedia class and the child “3DRoot” that is the root of the special 3D elements.

| <!– parameter entities –> |
| ENTITY % HFunctRoot "(seq?, *par*, layout, (%HFunctRoot;))?" > |
| <!– main elements –> |
| ELEMENT Harmonization (meta*, (%HFunctRoot), (Specialized)*) > |
| ELEMENT Specialized (Hobj3D, 3D-related-elements) > |
| ELEMENT Hobj3D (Common, 3DRoot;) > |
| ELEMENT Common (temporal*, spatial*, descriptor*, functionality*) > |
| ELEMENT 3DRoot (3D-related-elements) > |

4. XML HARMONIZATION SYSTEM

4.1. Harmonization architecture

The XML harmonization is achieved by the three-tier system as shown in Fig. 3. The bottom layer contains the \( N \) input XML applications to be harmonized, where each XML \( X_i \) has its corresponding virtual machine (VM) defined on the second layer. The VMs implement the harmonization transform from \( X_i \) to part of the \( X_{i+1} \) by the restructuring and integration operators in Eq. (2). The top tier is the user interface, which is the HXML. In this research, we use SMIL and X3D as input XML applications to demonstrate the proposed harmonization approach.

![Fig. 2. The OO-based harmonization data model.](image)

![Fig. 3. The harmonization architecture.](image)

Table 1. Partial HXML DTD.
4.3. Harmonization Algorithms

Two algorithms are designed for harmonization of DTDs and XML instances.

Harmonization of DTDs

The skeleton of the DTD of HXML is designed according to the analysis of the OO data model. After having the coarse HXML DTD structure, we can create detailed parts, i.e., the sub-DTDs that associated with each class. They are transformed and integrated from various XML applications and done by Virtual Machines (VMs). There are three steps in the DTD harmonization, which are done semi-automatically as described below.

Step 1: Determine which elements in the original XML applications belong to which classes in the proposed data model. The process is done by the analysis of users.

Step 2: The XML Virtual Machine extracts the determined elements and their attributes and transforms them into HXML sub-DTDs.

Step 3: Additional functionality rules are assigned to the corresponding HXML elements so that functionalities from different XML applications can be harmonized.

Harmonization of Instances

The harmonization of XML instances depends on the DTD of HXML. There are two scenarios. The first one is to transform the instances of original schemas into HXML instances. The second one is to create new HXML instances that conform to the DTD of the HXML. The transformation of XML instances can be summarized into the following three rules.

Rule 1: Elements in the “Meta” class are transformed to the sub-elements of the <Harmonization> element.

Rule 2: Elements belonging to the “Hfnct” class are preserved at the same position in the structure.

Rule 3: Elements belonging to a specific multimedia object is kept at the same position but all their attributes are checked. If the attributes have the corresponding functionalities, they are transformed to be object’s descendents.

5. CASE STUDY: HARMONIZATION OF SMIL AND X3D

We implemented the proposed harmonization system as shown in Fig. 3. SMIL and X3D were chosen to be the input XML applications. DTDs of SMIL and X3D were classified according to the OO data model given in Fig. 2. Let \( X_S = \{ F_S, O_S \} \) represent the SMIL and \( X_X = \{ F_X, O_X \} \) represent the X3D. The harmonization result can be represented by \( X'_{HXML} = \{ h_{11}(F_S) \cup h_{11}(F_X), h_{12}(O_S) \cup h_{12}(O_X) \} \).

We compared the OOHARMORIZATION approach with XMT which interacts with existing XMLs including SMIL and X3D. Let \( X_{SS} = \{ F_{SS}, O_{SS} \} \subset X_S \) and \( X_{XX} = \{ F_{XX}, O_{XX} \} \subset X_X \). We use \( X'_{XMT} \) to represent XMT and \( X'_{XMT} \) to represent the sub-schema of XMT that related to SMIL and X3D. We have \( X'_{XMT} = \{ X_{SS}, X_{XX} \} = \{ \{ F_{SS}, O_{SS} \}, \{ F_{XX}, O_{XX} \} \} \). The major difference between HXML and XMT lies in the structures of \( X_{HXML} \) and \( X'_{XMT} \). In HXML, all functions from SMIL and X3D are harmonized into the same set, and so are objects. In XMT, they are still separated.

The proposed OO-based harmonization can be generalized along three directions and will be reported in our later work. First, we can expand an existing harmonization work, i.e., to add new XML applications in to the HXML. At the implementation level, whenever a new XML application is to be harmonized, we can generate its corresponding VM to take care of the restructuring, transformation, and integration. The harmonization product, HXML, becomes \( X_{HXML} = \{ h_{11}(F_S) \cup h_{11}(F_X) \cup \cdots, h_{12}(O_S) \cup h_{12}(O_X) \cup \cdots \} \). Hence, the harmonization can be viewed as a process of building and enriching a library of functionalities and objects. Second, when a new concept is going to be added to an existing harmonization work, we can simply create new classes, which are expressed by new semantic sets in each \( X_i \). Third, one can create a new HXML for a different application domain. The provided OO data model, system structure, and algorithms can be implemented in a similar fashion.

6. CONCLUSION

The XML harmonization problem was formalized mathematically and a concrete OO approach was proposed to tackle this problem for applications in the multimedia domain in this research. The harmonization product, HXML, was designed to echo the OO data model, which accommodates functionalities and objects from various XML applications. The proposed OO harmonization methodology has great advantages in terms of extendibility and reusability.

7. REFERENCES


