A Survey to Conceptual Modeling for XML

Haitao Chen
College of Computer Sciences
Beijing University of Technology
Beijing, China
chenheyuzhi@yahoo.com.cn

Husheng Liao
College of Computer Sciences
Beijing University of Technology
Beijing, China
liaohs@bjut.edu.cn

Abstract—XML has become the de facto standard for representing and interchanging information in web-based applications. And conceptual modeling approaches, such as EER and UML, have been widely applied to illustrating data structure and relationships in traditional database. However, there is no standard methodology for designing XML data at conceptual level. In this paper, we give a survey to conceptual modeling for XML in terms of the basic models they employ. Most of them are based on well-established conceptual models. Some novel conceptual models are proposed to capture the hierarchical structure of XML. A set of requirements for a good conceptual models for XML are concluded. In particular, we emphasize the two phases modeling methodology and the importance of data semantics.

Keywords—conceptual model; XML; constraints

I. INTRODUCTION

XML has become the de facto standard for representing and interchanging information in web-based applications. Since more and more XML documents have emerged, native XML database has been designed and developed. The users can organize data based on XML data model directly. XML Schema, as a standard language for defining the schema of XML data, can be used as the logical schema of an XML model [1]. Typically, XML Schema is described by one or more XML files and lots of information is stored that is unrelated to applications (e.g. namespace, form). Therefore it is not intuitive for users to design the schema of XML data.

It is acknowledged that conceptual models are helpful for designing systems or databases. Conceptual models can capture the concepts and requirements of an application that are close to users’ intuitive perception. Thus, they can shorten the gap between end-users and designers and facilitate expression of applications. In the past few decades, ER model has been successfully applied to the design of relational database. UML is another widely used conceptual model for object-oriented analysis and design. However, there is no standard methodology for designing XML data at conceptual level. Although a conceptual model describes the concepts of real world, which is independent of the specific data organization, we believe that each conceptual model has its field of application. XML data model has its own characteristics (hierarchical structure, ordering etc.). Therefore, a new conceptual model should be devised.

Requirements for conceptual models for XML have been proposed. Most of them extend the existing conceptual models, such as ER, UML and ORM. However, they are not suitable for XML data because of XML-specific characteristics. A few new models are proposed for semi-structured data, which start from a hierarchical structure, such as ORA-SS [4]. They are not natural for modeling concepts, especially the objects with non-hierarchical structures.

In this paper, we first give a survey to conceptual modeling for XML. We put the previous work into five categories in terms of their basic models and most of them are based on well-established models. We discuss some critical problems and a set of requirements for a good conceptual models for XML are proposed. In particular, we emphasize the two phases modeling methodology and the importance of data semantics.

The rest of this paper is organized as follows. Section 2 surveys the previous work in conceptual modeling for XML. Section 3 discusses the problems and proposes a set of requirements. Section 4 concludes the paper with a brief discussion about future work.

II. PREVIOUS WORK

In this section, we survey the previous work, which are put into five categories in terms of the basic models they employ. Obviously, most of them are based on existing models, such as EER, UML and ORM. In what follows, we will use CM\(_i\) to denote the conceptual model proposed or employed in reference [i].

A. Conceptual model based on ORM/NIAM

1) CM\(_{ORM}\)

The use of ORM as a means for designing XML-schemas is investigated in [5]. The authors argue that the ORM can help designers get the ‘correct’ modeling. The role-based notation allows models to be easily validated with domain experts by natural verbalization and sample populations. ORM has rich expressive power, which can allow more business rules to be captured. To facilitate the process of modeling, an algorithm that can map an XML-Schema file into an ORM conceptual data model automatically is proposed. Although ORM can specify a wide variety of data constraints, including mandatory role, uniqueness, subset, exclusion, frequency and ring constraints, it fails to describe the integrity constraints, which play an important role in maintaining XML data. It is also hard to capture some XML-
specific characteristics, such as order and hierarchical structure.

2) CM$_{[6]}$

Reference [6] describes a semi-automatic process for converting an XML DTD to a schema in a canonical conceptual model based on ORM/NIAM and extended ER models. The CM$_{[6]}$ is not a new formalism for conceptual modeling, but an adapted mix of ORM/NIAM and EER to support semi-structured data representation. We classify it as ORM/NIAM-based model since the conceptual basis of it comes from ORM/NIAM. In particular, a composition relationship notation (an arrow from the composite to the component concept) is introduced, which is used to model the relationships in a hierarchical object. In a word, the CM$_{[6]}$ tries to make the best of ORM/NIAM and EER to capture the hierarchical feature of XML data. The main limitation is short of mechanisms to specify more data semantics, including the integrity constraints, ordering between data, etc.

3) CM$_{[7]}$ and CM$_{[8]}$

In [7], the NIAM is used as a conceptual model for the design of object databases. And the conceptual schema can also be transformed into an XML schema. The authors argue that it is a good approach to XML schema design since NIAM gives a conceptual framework for the design. NAIRM is a fact-oriented conceptual schema model, which uses reference type and fact type to express the relationships between data. It is natural to model the applications and the hierarchical structure. However, it can not describe more data semantics, such as the integrity constraints, XML-specific requirements. Reference [8], whose major objective is to reengineer poorly designed XML Schemas into the normalized ones, also employs the NIAM as the conceptual schema model of XML databases.

4) CM$_{[9]}$

In [9], an XML conceptual schema is presented using the ORM conceptual model. The authors focus on constraints and derivation rules. It is argued that although most constraints can be neatly represented on conceptual schema constructs, some constraint (e.g. dynamic constraints) need to be represented in other ways (e.g. by logical formula or program code). And derivation rules provides a list of functions, operators and rules that are used to derive information. The constraints that can be expressed in ORM and XML Schema are concluded respectively. However, some constraints that can be specified in XML Schema can not be modeled by ORM. More importantly, more integrity constraints, such as functional dependencies, multi-valued dependencies and inclusion dependencies, can not be specified in a standard way. That means a conceptual model should take into account more integrity constraints and make its expressive power more. XML-specific characteristics also need paying more attention.

B. Conceptual model based ER

1) CM$_{[10,13]}$

ERX is proposed in [10], which is an evolution of ER and can cope with the peculiar features of XML. ERX is not based on nested structures and links among them: it exploits a flat representation to put in evidence XML concepts and relationships among concepts. ERX provides the concept of Hierarchy to capture the hierarchical structure of XML data. In relationships, order and cardinality constraint are considered. The attributes are associated to a qualifier, which denotes the attribute is required or optional. It is argued that existing conceptual models are not suitable to semi-structured data in [11]. Then the authors propose a set of minimal extensions to ER that try to capture information in an XML DTD. Specifically, all attributes in ER are marked as optional or required and a choice attribute is represented explicitly. Another model, XER, is proposed in [12], which is also an extension of ER. The entities in [12] have three types: ordered entity, unordered entity and mixed entity, which are used to capture the order and mixed content in XML data. XER also provides the concept of generalization to denote an IS A relationship. ErX is presented in [13], which is similar to [12]. The IS A relationship is described by a structural specification named categories. The coverage and order constraints are also considered.

They are the early attempts for extending ER with abilities to describe XML-specific features. However, it seems to be not enough, since XML Schema has become more popular, which has richer expressive power than DTD or XGrammar Schema. There are more data semantics that should be captured, which are even can not be described by XML Schema, such as the integrity constraints. And ER is not hierarchical and there is a problem with modeling of a hierarchical structure of XML data.

2) CM$_{[14]}$ , CM$_{[15]}$ AND CM$_{[16]}$

A methodology is designed to map a relational database to an XML database for database interoperability in [14]. The relational database is transformed into EER model. Then the EER model is mapped to XML schema. EER is employed as a conceptual model to connect two different data models. A similar work can be found from [15], which focuses on preserving integrity constraints. An interested issue is discussed in [16], that is, because XML is hierarchical, many different XML schemas can be generated from the same conceptual schema. EER is employ as the conceptual model and relational schemas can be reverse engineered into EER schemas. Users can easily customize XML document structures for their specific applications by interaction with system. They are reverse engineering and the ER model is a bridge. It is not a methodology for designing XML by EER. Also, there is no necessary extension for supporting XML-specific characteristics.

3) CM$_{[17]}$

A conceptual model, named XSEM is proposed in [17]. The main idea is to divide the XML conceptual modeling process into two parts. The first part consists of designing an overall conceptual schema using an extension of ER called XSEM-ER. The second part consists of designing a hierarchical organization of the structure from the first part using a hierarchical model called XSEM-H. It is also argued that there may be more than one hierarchical organization of the same data modeled by an XSEM-ER schema. Data node types and cluster types are proposed for modeling semi-structured, irregular and heterogeneous data. It is a novel idea to divide the modeling process into two phases. We
believe that it is reasonable to distinguish these two phases for modeling XML data. In the first phase, users can focus on modeling the logical structure of data. And in the second phase, users can decide how to store data. However, there is no more discussion about data semantics.

C. Conceptual model based on UML

1) CM\[18\]

It is an early work for representing XML schema by using UML. The authors have used the UML extension mechanisms (stereotypes and tagged values) to create new classes of UML objects to explicitly represent XML artifacts. Specifically, all element and data types in XML Schema are mapped to classes annotated with stereotypes, which reflect the semantics of the related XML Schema concept, such as sequence group or a required attribute. Special stereotypes are introduced to indicate a class represents an anonymous group of elements. Ordering is described by a sequence number. It seems to be not natural to use UML to express the hierarchical XML data. And there is no discussion on how to use UML constructs to express identity constraints.

2) CM\[19\]

UML’s Static View and Model Management View are used to represent the XML data at conceptual level. Specifically, they consist of Class diagram and Package Diagram. To take advantage of all facets that DTD concepts offer, necessary extensions to UML have been made, which are UML compliant and as minimal as possible. Some class and attribute stereotypes are introduced to express the concepts in DTD, such as entity. Comment ‘sequence’ is used to model a sequence explicitly. The ‘choice’ constraint is employed to describe alternative element contents. The mixed content can be specified by ‘choice’ constraint with one occurrence of #PCDATA. A ‘discriminator’ constraint is introduced to express the generalization specification. It has the same disadvantages to [18].

3) CM\[20-22\]

Object-oriented conceptual model (e.g., UML) is used as the conceptual model for XML in [20]. The authors discuss the transformation from the OO conceptual models into XML Schema, with a focus on the transformation of aggregation and generalization relationships. UML is extended in [21] to represent XML Schema at the conceptual level. The extension defines a set of stereotypes, tagged values and constraints that enable users to represent in graphical notation in UML all the components of an XML Schema, such as <<Choice>> and <<Sequence>>. A conceptual model, UXS (UML & XML Schema), is proposed in [22], which is based on UML and provides several graphical constructs to help the users to define complex XML schemata. In particular, the UML is extended with suitable stereotypes according to the components defined in the XML Schema. For example, the occurrence cardinality can be specified. There are also corresponding constructs for elements, attributes, simple types, complex types, content models, key constraints and notations. The mechanisms for including external schemata are also considered. The above methods are based on the extensive mechanisms in UML, which usually make a one-to-one and semantically equal mapping between UML constructs and components in XML schema. It is easy to understand and use. However, more semantics need considering. And we also need a better way to express the hierarchical structure.

4) CM\[23\] and CM\[24\]

A three level design methodology is presented in [23]. The conceptual level model is to model the domain using UML class diagram with some non-standard annotations that is used to represent some common conceptual constraints, such as the primary identification of a class. The logical level model is a direct representation of the XML Schema, which describes the physical data structure in an abstract way. The physical level model is the textual representation of XML Schema. An approach for reverse engineering XML to a conceptual model by using UML is presented in [24]. Similarly, a three-level model for reverse engineering XML DTD and Schema is defined. The concepts in DTD or XML Schema are captured by using UML extensive mechanism. These two researches both focus on the design methodology. The three level approaches are inspired by the traditional database design methodology. UML are employed as the conceptual model with some necessary extensions. They have similar limitations to other UML-based models.

5) CM\[25\]

The UML is employed in [25] to facilitate understanding of XML Schema. The thirteen XML building blocks (such as element declaration, complex type definition and identity constraint definition) defined in the XML Schema can be mapped to the UML constructs by applying a set of transformation rules. The main different is there is no new stereotype, which can simplify the resultant UML diagram and make it easier to understand. We argue that it seems not to be intuitive for a conceptual model to describe some constraints. Also, there are more constraints that can not be specified by XML Schema so far. A conceptual model should have mechanisms to express them.

D. Conceptual models based on tree structure or graph

For capturing the hierarchical structure of XML, some new models are proposed, which are based on tree structure or graph. Intuitively, these models can represent the hierarchical structure naturally. However, it is not a reasonable way to model application at conceptual level, since it exposes more details on organization of data. It is also unnatural to model data that is not hierarchical structure.

1) CM\[4\]

The Object-Relationship-Attribute model for SemiStructured data (ORA-SS) is propose in [4], which consists of three basic concepts: object classes, relationship types and attributes. An object class is similar to en entity type in ER. A relationship type describes a relationship among object classes. Attributes are properties that belong to an object class or a relationship type. The relationship type and attributes that associate with one object class are denoted by an arrow, which can show the nested structure of semi-structured data. Some constraints such as occurrence cardinality, key and foreign key can also be expressed in ORA-SS. Although it is easy to model hierarchical structure, the abstract level seems to be not so high, since designer has
to consider the data structure. It is also short of discussions on more data semantics. And the problem arises when we try to model non-hierarchical data model.  

2) $CM_{26}$, $CM_{27}$ and $CM_{31}$  

XML Schema Definition Language (XSD) Graph is presented in [26], which can show the data semantics in a more user-friendly approach. More semantics are considered, such as functional dependency, inclusion dependency and multi-valued dependency. XSD Graph is a graphical XML Schema with rich data semantics. A similar approach is presented in [27], which employ the DTD Graph as the conceptual schema, that is, the logical schema in [27] is described by DTD. An XML Tree Model (XTM) is proposed in [1] as an XML conceptual schema for representing data semantics in a diagram. An XTM consists of hierarchical nodes representing all the elements and the data relationships among elements within the XSD. Some data semantics can be captured by XTM, such as cardinality, generalization. The main disadvantage of these methods is the exposure of data structure, which makes the abstract level low. Strictly speaking, they are not suitable to be used as the conceptual model, since a higher abstraction is desired.  

3) $CM_{29}$  

A new modeling language, Conceptual CML (C-XML) is proposed in [28], which is based on hypergraph and has three concepts: object sets, relationship sets and constraints over these object and relationship sets. Participation constraints can be specified to show how many times an object in a connected relationship may participate in a relationship set. Optional or mandatory participation can be defined. C-XML emphasizes some XML-specific components in XML Schema, including sequence, choice, mixed content, any and anyAttribute. The C-XML can represent the hierarchical structure of XML data naturally. It can also describe rich data semantics and XML-specific characteristics. However, it is also unnatural to model non-hierarchical data model. In other words, the abstract level seems to be not high, which makes it not suitable to model applications.  

E. Conceptual models based on Semantic net  

XML Semantic (XSemantic) net is a modified semantic network model notation to model XML domains using the OO concept, which can capture most XML-specific constraints because of its structural similarity to XML [29]. In [30], an eXtensible semantic (XSemantic) net is presented, which has a special type of node, i.e. event node that is used to capture the dynamic properties. We do not go into detail in this paper. The main disadvantage is there is no sufficient support for rich data semantics.  

III. DISCUSSION AND REQUIREMENTS  

A. Discussion  

1) The conflict between conceptual modeling and data organization  

There is a conflict when we try to model the XML data at conceptual level. On the one hand, the conceptual modeling requires a high level abstraction, which means the details for data organization should not be exposed. On the other hand, if we want to capture XML-specific in conceptual model, we have to reveal some features on data organization. A feasible way is to divide the modeling process into two phases like the proposal in [17]. The concepts of applications are modeled in the first phase and the data organization is decided in the second phase.  

2) Based on existing models or a novel one  

We can find that most of existing investigations are based on well-established models, such as EER, UML and ORM. It is easier to use and understand them. For several novel proposals, they have the similar basic ideas with a focus on the XML-specific characteristics, such as the hierarchical structure. It is not important to employ model that are based on well-established model or a novel one. The important thing is they should be with high abstraction level and graphical notations. They are also required to have formal foundations and be independent on XML schema language.  

3) The role of data semantics  

Data semantics play an important role in making the data more meaningful and shareable. We emphasize that it is very useful to capture richer data semantics at conceptual level, which can model the applications more accurately. The previous proposals can model data semantics more or less. However, we should pay more attention on how to support more data semantics, especially, the integrity constraints for XML, which have been widely studied for the past few years [31]. Unfortunately, there is no standard way to define more integrity constraints, such as functional dependencies, inclusion dependencies and multi-valued dependencies. It deserves further investigation. How to express these constraints in graphical notations is also an interesting topic.  

B. Requirements  

We conclude a set of requirements for a good conceptual model for XML data, which is similar to [2] and [3], but with a focus on the two phases modeling methodology and data semantics.  

1) Independent on XML schema language  

Although we are interested in developing a conceptual model for XML and XML Schema seems to be a promising XML schema language, the conceptual model should have high abstraction level.  

2) Formal foundations  

The model should be developed based on a formal foundation, which is useful in analysis of the properties of model and facilitate related applications, such as data transformation.  

3) Graphical notations  

It is a common requirement, which can make the model user-friendly. And a design tool is desired to help users model the applications.  

4) Relationship types  

To model the applications, one important aspect is to capture the relationships between concepts. A good model should provide mechanisms to model N-ary relationship types, to describe the attributes of relationship types. And there should have mechanisms to express some special relationship types, such as aggregation, generalization.
5) Data semantics

A good model should have sufficient mechanisms to describe data semantics. One important kind of data semantics is the integrity constraints. We emphasize that to integrate the integrity constraints into conceptual model is important for the applications. How to express the integrity constraints in a graphical way is a difficult issue, since we have to consider the simplicity of the model.

6) XML-specific characteristics

XML Schema is often used as a logical representation for XML data. For translating the conceptual schemas into XML Schema, we must tackle some XML-specific characteristics, such as namespace. For preserving the high abstraction level, a two phases modeling methodology is a good choice.

IV. CONCLUSION AND FUTURE WORK

Conceptual modeling for XML is fundamental to design XML data and related applications. It is important to have a standard methodology. We survey the previous work and discuss some related issues. Some requirements are proposed for a good conceptual model for XML. We believe a two phases modeling methodology is a reasonable approach and the data semantics are critical for conceptual models.

We are currently working on developing a new conceptual model for XML, which employ the two phases modeling methodology. For future work, we are interested in making our model more powerful. Especially, we are integrating more integrity constraints into our model. How to apply our model also deserves investigation.

ACKNOWLEDGMENT

This research is supported by Beijing Municipal Natural Science Foundation (No. 4082003).

REFERENCES


