Special Issue on Research and Innovations in Science and Engineering
Volume: 5 Issue: 12s
ISSN: 2454 - 4248
164 - 169

XML-based Domain Modeling using Xindice

Dr. Prasanna M. Hasabnis
Assistant Professor
Dept. of Computer Science & Engineering
Mauli Group of Institution's COE & T, Shegaon
ail: prasanna.hasabnis23@gmail.com

Dr. M.S. Ali
Principal
PRMCEAM, Badnera
email: softalis@gmail.com

Abstract - This paper describes the technique of knowledge representation for a subject domain. After a brief introduction, the features of XML are described. Next section describes the proposed model of XML based subject domain. Next section describes the introduction of XML database server Xindice and its various features. Finally, the last section concludes with the conclusion.

Keywords - *domain modeling, XML, Xindice*

*****

I. INTRODUCTION

The important role that knowledge plays in building intelligent systems is now widely accepted by practitioners in AI. Knowledge can also be defined as the body of facts and principles accumulated by human kind or the act, fact, or state of knowing. The meaning of knowledge is closely related to the significance of intelligence. Intelligence requires the possession of knowledge and access to it. And characteristic of intelligent people is that they possess much knowledge [Patterson, 2002]. Knowledge representation developed as a branch of artificial intelligence in general and expert systems in particular. One of the components of expert system called 'knowledge base' needs an effective method of knowledge representation for the domain of expertise so that the 'inference engine' can use it for reasoning and problem-solving phase [Rolston, 1988]. Knowledge representation has a long tradition rooted mainly in philosophy, logic, and psychology. It is the application of logic and ontology to the task of constructing computable models for some domain.

According to Davis [Davis et al.,1993], there are five basic principles about knowledge representation –

- A knowledge representation is a surrogate.
- A knowledge representation is a set of ontological commitments.
- A knowledge representation is a fragmentary theory of intelligent reasoning.
- A knowledge representation is a medium for efficient computation.
- A knowledge representation is a medium of human expression.

In order for the system to have the versatility that the teaching-learning process requires, the representation of the instructional knowledge should follow some guidelines

[Redondo, 2006]:

- Structured - The fundamental factor of an intelligent system is its knowledge structure. The codification must be such that it allows to increase or to decrease the structuring degree according to the author and the needs of the system. This is normally done through entity-relation hierarchies.
- • Characterized - The relationships that define the internal structure must contain parameters that facilitate the adaptability of the process. Since the objective is to transfer a piece of knowledge, the internal description will be centered in pedagogical aspects like Depth Level (Memory /

ISSN: 2454 - 4248
164 - 169

Understanding /Application), Type of Transference

(Generalization / Explanation, Classification, Assessment), etc.

- Categorized - Instructional knowledge must follow a hierarchy of categories or classes that define their structural logic (Fact / Concept / Procedure, Subject/Section, etc.).
- Coherent - Each element of the structure must be independent and complete, so that the system may be able to explain a complete course or to respond to smaller queries.
- Continuous - Entities that form the knowledge must be joined by means of some type of glue knowledge [Reigeluth, 1987] that facilitates a continuous instruction.
- Regulated - Instructional knowledge must incorporate the experience that facilitates its sequencing. Normally, different ways of presentation are included, such as tracking elements (queries and tests) and even erroneous knowledge [Ohlsson, 1986], with the aim to guide the student and to avoid the most common errors.

II. STRUCTURE OF DOMAIN MODEL

Domain modeling can be based on the premise that learning resources in conventional educational set up are organized as a hierarchy. Learning resources are divided into many subjects according to the institution. One subject is divided into many units; one unit is divided into many atomic materials and examinations. Once this hierarchy is defined, sequence is automatically determined for the subjects, units, topics down to the materials and tests. This hierarchy is followed in top-down manner during design phase, while it is used in bottom up fashion during execution phase. However, learning activity will not complete only with the transfer of these materials. Skills of absorbing, recalling, joining, applying and selecting learnt information are the other important issues in learning activity. These skills are called 'learning strategy'. Learning strategy is domain dependent, but some strategy is reusable to other domains. Moreover, navigational skill of teachers is the part of information of learning activities. Skill of selecting, joining, communicating and emphasizing the learning material is termed as “instructional strategy”. This strategy is thought to be relevant to the learning strategy, and also be domain dependent. This instructional strategy can be implemented as some specific framework of navigation. SCORM specification [SCORM, 2004] has a straightforward and simple navigation flow, but it will be more flexible to reflect the navigation framework based on both learning and instructional strategies. When these strategies are identified as a part of a domain model, domain modeling will have a new way to accelerate its productivity and quality. Even if learning materials of different subjects are developed, the navigation and feedback framework can be reused if these subjects share the common learning or instructional strategy. This is the basic theme used in this research for domain modeling.

Specifically, the representation of a subject domain knowledge is motivated from the lesson planning activity of a human tutor. A human tutor plans her teaching activity from the curriculum, the learning objectives, the time duration, the learner’s background and the available course material. Although teaching strategy may differ from subject to subject, a common thread can be identified in planning activity. To truly mimic the behavior of a human tutor, her lesson plans serve as good guidelines for domain knowledge representation. Using this framework of lesson planning by a human teacher, following requirements for domain modeling are identified:

i. Subject independence,
ii. Simplicity,
iii. Ease of dynamic lesson generation, and
iv. Ease of lesson sequencing and presentation.

III. REVIEW OF XML

[Bradley, 2000; Goldfarb & Prescod, 2003; Zeid & Gupta 2001; Young,2001] XML or eXtensible Markup Language is a text based markup language that allows storing data in meaningful and structured way. It is extensible, allowing users to define their own tags with their own attributes and values. These values offer semantic qualifications to data and context. XML provides context validation at both the semantic and syntactical levels. It checks for completeness and well-formed ness of documents. XML also supports rich

ISSN: 2454 - 4248
164 - 169

structures, similar to those found in object oriented programming, and database applications. XML is a meta language. XML is simply a set of rules for creating new document types. XML defines XML documents. XML documents define data objects. XML is constructed on the well-known object oriented paradigm. XML documents offer access to the content and structure of the data objects. An XML document uses tags to define both the data and structure of the objects that the document defines, and eventually creates. The structure of the data defines the object hierarchy. XML tags can be nested. They may have attributes. Attributes may have values. XML tags follow strict rules. They are case sensitive, white space matters, and all attribute values must be enclosed in quotes. The logical structure of an XML document consists of declarations, tags (elements), comments, characters, and processing instructions. All of this logical structure is specified in the document by explicit markup. The logical structure of an XML document refers to its tags, their structures relative to each other, and their semantics. Tags come in two groups. One group has tags that start and end. The other group has empty tags. The logical structure must be well formed and valid. During the processing of XML documents, the wellformedness and validity requirements are used by XML parsers (processors). It is much easier to check the wellformedness of a document than checking its validity. All what we need to do to check if a document is well formed, is to check its tags against XML syntax rules. Checking the document validity requires us to establish the semantics rules that we use as a reference to measure the document semantics against. These semantics are included in the DTD (document type definition). Thus, we need to write a DTD for every XML application we create. XML parsers validate XML documents using the DTDs. A DTD contains statements that define to the parser what is possible to do in a valid XML document that uses this DTD. Each statement looks like a tag. XML parsers

read all DTD statements before they begin parsing a document. XML documents could be referring to other documents that they may need. The physical structure of an XML document consists of all entities that are contained in the document. The main document represents the "root" or the "document" entity. The document entity serves as the starting point for an XML parser. Entities could be parsed or not. Parsed entities are replaced by their corresponding text that becomes part of the main XML document. Unparsed entities usually refer to a resource whose contents are not text. For example, an executable that is referenced in an XML document is an unparsed entity.

IV. XML BASED DOMAIN MODELING

The idea behind domain knowledge representation in the proposed ITS is to have–to the most possible extent-a subject independent domain representation. The subject independence nature will help in designing the domain independent tutor module Further, the representation should facilitate 'adaptivity' of lesson content according to the learner's knowledge level. The sequencing and presentation of these lessons to the learner are the other issues which can be sorted out easily if the representation is effective.

A. Efficacy-based Content Classification

One of the most important features of the proposed domain modeling language is the 'content classification'. While all of the learning material markup languages or the meta-data specifications do have content classification, it is either based on the content type or the learning goal. None of the specifications, to the best of our knowledge, points out the efficacy of the content to enhance a particular cognitive skill of the learner. We advocate the Bloom's opinion [Bloom et al., 1964] that the memorizing ability, the understandability, the concept comprehension and even the misconception are the cognitive skills that can serve as better indicators of learner's progress. Bloom's taxonomy is learner-focused dealing with the types of learning viz. cognitive, affective and psychomotor meaning the knowledge level, the attitude and the skills respectively [Bloom et al., 1964]. However, it provides certain guidelines for instructional design, more specifically guidelines for test design. Based on these guidelines we opine that for achieving 'adaptivity' the content should be classified on the basis of their efficacy to strengthen learner's cognitive level. Consider, for example, one category in cognitive domain of the taxonomy named as 'Knowledge: Recall of data'. The examples cited indicate that after a learning experience, the learner can be considered to have acquired knowledge if he can "Recite a policy. Quote prices from memory to a customer. Knows the safety rules". The keywords at this level are: defines, describes, identifies, knows, labels, lists, matches, names, outlines, recalls, recognizes, reproduces, selects, states. If these verbs could be used to identify the cognitive level of the learner, they should be usable to classify the learning material. Obviously, this demands much more efforts and expertise on the part of the instruction designer or the content author. The cognitive domain of the taxonomy itself enumerates as many as ninety eight keywords! The content classification at such a micro-level may turn into a futile effort. To remedy this situation, we propose a simple classification based on learner's observable response. On the basis of efficacy of the content to strengthen memory, understanding and / or conception twenty three categories are identified.

B. A Three Layer Domain Model

The proposed architecture for a three-layer domain model consists of a user layer, a conceptual layer and a physical layer. The functionality of each of the layers is described below.

User-Layer: It consists of Lesson Objects (LO) for the presentation of course material to the learner. This is similar to the Sharable Content Object (SCO) as defined in SCORM [SCORM, 2004]. However, its structure is simpler than SCO, and it is obtained after aggregation. It is interfaced with the user through eXtensible Stylesheet Language (XSL).

Conceptual Layer: A meta model that describes a hierarchical structure of the domain. It is the main concern of

this chapter. It is similar to the Content Organization as defined in SCORM. It is a map that represents the intended use of the content through structured units, topics and the lessons. It is this layer that is referred by the tutor module for sequencing the lessons, topics and the units. The relation between a lesson and a lesson object is similar to the relation between a class and object in object-oriented-programming paradigm. While a lesson is an abstract idea, a lesson object (LSO) is a concrete entity composed from the learning assets according to the lesson plan.

Physical Layer: Actual content consisting of text, images, figures, code, facts, principles etc. named as 'Assets' in SCORM. It is the most basic form of a learning resource. Assets are nothing but the e-representation of media, such as text, images, sound, assessment objects or any other piece of

data that can be rendered by a Web client and offered to a learner. Assets are stored in XML database and can be updated through XUpdate interface. The retrieval of the assets for aggregation can be done using XQuery language or XPath.

The subject domain is composed of several Units. Each unit

is composed of several Topics. Each topic being the combination of several Lessons. Thus, a hierarchical structure very similar to a textbook is designed, wherein a unit is equivalent to a chapter, a topic is equivalent to a section and a lesson is similar to a subsection. A lesson is composed of description of the concept with examples and explanation. The physical layer consists of concept description, examples and explanation. The objects at the physical layer are designed in such a way that they can be reused. The objects at the conceptual layer are the units, topics, and the lessons. The course generator generates lesson objects dynamically for each concept of a topic in a unit. The number of lessons for a topic, the number of topics for a unit and the number of units for a subject domain are fixed by the curriculum designer.

C. Domain Model Implementation

The capability of the rapid web development architectures that may be used for education is given below.

- Dynamic selection of content (through Xpath, XQuery support) and logic sheets through a web interface (through XSL support) to control the display of
- content to meet the needs of different audiences.
- Storage of information by descriptive metadata making it (through NativeXML Databases) searchable and reusable.

ISSN: 2454 - 4248
164 - 169

- Storage of content stored in an XML language based on the domain ontology (through XTM).
- Web access to resources, databases and xml-files (through XPath).
- Web Forms and Services (through XUpdate) for easy upload to the server.

Our work is mainly concerned with the first two type of interfaces; there too we only demonstrate the technical feasibility of the proposed model. For identifying a support tool for domain model feasibility, we carried out an exhaustive search for open source tools and found out a native XML database system named as Xindice [Source-Forge, 2005].

D. Features of Xindice

1. Document Collections: Documents are stored in collections that can be queried as a whole. We can create collections that contain just documents of the same type or we can create a collection to store all our documents together. The database doesn't care.
2. XPath Query Engine: To query the Document Collections we use XPath as defined by the W3C. This offers a practically flexible mechanism for querying documents by navigating and restricting the result tree that is returned.
3. XML Indexing: In order to improve the performance of queries over large numbers of documents we can define indexes on element and attribute values. This can dramaticallyspeed up query response time.
4. XML:DB XUpdate Implementation: When we store XML in the database we may want to be able to change that data without retrieving the entire document. XUpdate is the mechanism to use when we want to do server side updates ofthe data. It is an XML based language for specifying XML revisions and allows those alteration to be applied to entire document collections as well as single documents.
5. Java XML:DB API Implementation: For Java programmers Xindice provides an implementation of the XML:DB API. This API is envisioned to bring portability to XML database applications just as JDBC has done for relational databases. Most applications developed for Xindice will use the XML:DB API.
6. Command Line Management Tools: To aid the administrator Xindice provides a full suite of command line driven management tools. Just about everything we can do through the XML:DB API can also be done from the command line.
7. Modular Architecture: The Xindice server is constructedin a very modular manner. This makes it easier to add and remove components to modify the server to a particular environment or to embed it into another application.

V. CONCLUSION

This paper has first introduced the concept of knowledge representation as applied in expert systems and showed how domain knowledge in the context of ITS differs that from artificial intelligence field. A brief review of knowledge representation techniques has been presented bringing out the shortcomings of the available metadata specifications. A new domain markup language i.e. XML is presented and its basic features are demonstrated. The three-layer architecture supports the principle of separation of concerns and facilitates multiple views of the same learning object.

We present below our explanation regarding the proposed approach.

i. How are the human characteristics incorporated in the model?

The domain model basically represents the curriculum portion of the teacher curriculum-learner triangle. It is a static entity to be used intelligently by the other humanistic components. Thus, directly there are no apparent human characteristics in the model except for the content organization at the physical layer.

ii. How is the ‘adaptivity’ achieved?

The adaptivity is achieved through the introduction of ‘content class’ that reflects the effectiveness of a particular learning object towards the enhancement of the learners cognitive level. As long as the performance remains at a particular level, the same lesson with different content can be presented. The

ISSN: 2454 - 4248
164 - 169

selection of lesson content becomes adaptive to the learner's knowledge level.

iii. Is the 'adaptivity' learner-centered?

As pointed out earlier, the material is organised according to its class which is selected on the basis of the learner's current performance class and hence the model is learner-centric.

iv. How is the 'intelligence' incorporated?

It is incorporated indirectly as a part of the instructional design when a class is assigned to a content.

v. Which of the learning theories advocate the model?

Contructivistic learning theory and cognitivistic learning theory advocate the model. Blooms taxonomy also supports such models.

vi. How is the proposed model domain independent?

The conceptual layer is subject domain independent. The physical layer cannot be subject independent because of ontology. However, no assumption is made about the subject domain nor any subject dependent parameter is incorporated in the model. As long as the subject expert can properly organize the content in a 'class', the given model can be used for any domain.

vii. What are the envisaged limitations?

The model demands huge efforts for classification on the part of instruction designer.

REFERENCES

[1] [Davis et al., 1993] Davis et al., "What is a knowledge representation", AI Magazine, 14(1):17-33, 1993. Available online: http://medg.lcs.mit.edu/ftp/psz/k-rep.html.

[2] [Rolston, 1988] Rolston David W., "Principles of Artificial Intelligence & Expert Systems Development". McGraw-Hill International Student Edition, ISBN 0-07-100283-9, 1988.

[3] [Redondo, 2006] Redondo C.A., "Knowledge Representation in an Adaptive Pedagogical Module". Current Developments in Technology-Assisted Education, 2006.

[4] [Reigeluth, 1987] Reigeluth C.M., "Instructional Theories in Action: Lessons Illustrating Selected Theories and Models". Laurence Erlbaum Associates, Hillsdale, New Jersey, 1987.

[5] [Ohlsson, 1986] Ohlsson S., "Some Principles of Intelligent Tutoring". Instructional Science, 14, 1986.

[6] [Bradley, 2000] Bradley N., "The XML companion", Addison- Wesley, 2000.

[7] [Goldfarb & Prescod, 2003] Goldfarb Charles F. and Prescod P., "The XML Handbook", Pearson Education, 2003.

[8] [Zeid & Gupta, 2001] Zeid I. and Gupta S. M., "Disassembly knowledge representation via XML", Proceedings of SPIE 4193- 179, 2001.

[9] [Young, 2001] Young M., "Step by step XML". Prentice Hall of India, New Delhi publication, 2001.

[10] [Bloom et al., 1964] Bloom B. S., Bertram B., Mesia B. B. and Krathwohl David R., "Taxonomy of Educational Objectives" (Two Vols: The Affective Domain & the Cognitive Domain). New York. David McKay, 1964.

[11] [Source-Forge, 2005] Source-Forge, accessible at http://www.sourse-forge.net/, 2005.