# DESIGN OF A SCHEMA-DRIVEN ENVIRONMENT FOR FACILITATING APPLICATION OF CONSTRUCTION INFORMATION STANDARDS

Hsien-Tang Lin
Department of Civil Engineering
National Taiwan University
Taipei 10617, Taiwan
nicep@caece.net

Shang-Hsien Hsieh
Department of Civil Engineering
National Taiwan University
Taipei 10617, Taiwan
shhsieh@ntu.edu.tw

Abstract: This paper presents the design of an XML schema-driven, tool-building environment, named XsEnvironment, to facilitate application of construction information standards. XsEnvironment is designed to take an XML Schema of a construction information standard, as input information, and then automatically build the corresponding form-based XML information processing applications, which may include XML information editor, database, and Web Services components. The information processed by the applications created in XsEnvironment can be saved into a valid XML file or a database. Furthermore, the information can be shared and exchanged through the web services provided by the XsEnvironment application tools.

Keywords: XML Schema, Construction Information Standards, Information Sharing and Exchange

## 1. INTORDUCTION

In recent years, increasing efforts on information standardization have been made for facilitating information sharing and exchange in many industries, including construction industry. The promotion of Construction CALS (Continuous Acquisition and Lifecycle Support) [1] may represent the early effort by the construction industry. Moreover, all of these efforts on information standardization employ XML (eXtensible Markup Language) technologies [2], in particular, the XML Schema [3], which provides a means for defining the structure, content, and semantics of XML documents. For example, ebXML [4] has been proposed to enable enterprises of any size, in any location, to share and exchange business information through Internet. aecXML [5] is being developed to standardize shared vocabularies in the Architecture, Engineering, and Construction (AEC) industry.

With the information standards defined in the format of XML schemas, software tools with user-friendly GUI (Graphical User Interface) are needed for people to easily digitalize and edit information in the format of XML files that conform to the required XML schema. To address this need, several researches and commercial tools have been proposed. For example, the commercial application, called InfoPath [6], is included in Microsoft Office 2003 to facilitate creation of form-based GUI for editing the XML-based documents. InfoPath provides a form designer for users to build the mapping between a tag in the input XML schema and a windows control. Although InfoPath is a general and flexible tool, the user is required to have some knowledge of the tags in the XML schema and their mapping to appropriate GUI components, and repetitive similar works are needed for building forms for different XML schemas. To automate the repetitive form-based GUI building tasks for different XML schemas, an XML information processor, named XsForm [7-8], was proposed by the authors. XsForm takes the user-input XML schema then automatically generate appropriate Windows GUI forms and associated functions for editing and management of XML information conforming to the input XML schema.

Furthermore, to facilitate application of information standards for information sharing and exchange, we do not only need XML information editing tools (e.g., XsForm), but also software tools for managing XML information using database technology and for communicating XML information through Internet. In this paper, an XML schema-driven, tool-building environment, called XsEnvironment, is designed for addressing the aforementioned needs in construction industry. After taking an XML schema (of a construction information standard) as input, XsEnvironment can automatically build the corresponding software tools for editing, management, and sharing of XML information conforming to the input XML schema.

The remaining of this paper is organized as follows. Section 2 analyzes the functional requirements of XsEnvironment using some possible scenarios. The system design of XsEnvironment is discussed in Section 3. Although XsEnvironment is still being implemented at this time of writing (June 2006), how it can facilitate application of information standards in construction industry is also illustrated in Section 3. Finally, Section 4 draws some conclusions.

## 2. SYSTEM ANALYSIS

To analyze the functional requirements for XsEnvironment, possible scenarios for application of tools provided by XsEnvironment in construction industry are discussed and analyzed. Then, the system requirements for XsEnvironment are summarized.

### 2.1 Application Scenarios

Two possible scenarios are discussed here for a construction company to deal with information standards expressed by XML schemas:

a. Preparing XML documents based on various information standards

Increasing number of construction information standards has been published for sharing and exchanging the e-business information between governmental organizations and AEC companies especially in the area of public work construction. To ease the tasks for preparing XML information for various XML schema based information standards, XsEnvironment can provide an effective and efficient solution by automatically generating form-based XML information processor according to the desired XML schema. This can avoid repetitive and costly development of similar editing tools for a large number of XML schema based standards. Also, from the user's point of view, the application of the editing tools automatically generated by XsEnvironment are convenient and user-friendly because they are very similar and with consistent look and feel. Therefore, a construction company should be able to employ XsEnvironment for preparing information for all kinds of standardized documents involved in bidding and executing public construction projects.

b. Gathering and sharing of construction information

A construction company may have several projects running at different construction sites. For the purpose of construction management, all construction sites are usually required to prepare and submit their construction daily reports to the company. Assume that the company has standardized the construction daily report in terms of an XML schema to facilitate the collection and sharing of daily construction information. For editing of the construction daily reports on portable notebooks or desktop computers at the construction sites and, XsEnvironment can be employed to generate a form-based construction daily report editor from the standardized daily report schema. For better management of large number of construction daily reports, the editor is equipped with a database that is also created automatically based on the daily report schema. In addition, the editor serves as a client program to allow for uploading and downloading information of daily reports to and from the database server located at the headquarter of the company through Web Services [9]. On the other hand, for collection of daily reports from construction sites, the company headquarter can employ XsEnvironment to automatically build a database server application on the server computer of the company based on the daily report schema. The database server application provides information retrieval and update services with authentication and authorization controls through Web Services.

Although the above scenario uses the construction daily reports as an example to discuss the possible application of XsEnvironment for gathering and sharing of construction information, it should be noted that, similar scenarios also apply to sharing and exchanging of other construction information within an AEC company or among business partners in the construction industry.

### 2.2 System requirements

After analyzing the possible scenarios in the previous section, we can summarize the major requirements for XsEnvironment as follows:

- The system (or environment) should be able to automatically build an appropriate XML information processor with form-based GUI based on the user-input XML schema. In addition, the XML information processor should allow for input of a new schema at any time and rebuild its form-based GUI accordingly and automatically. Functions for editing, loading, saving, and managing XML files should also be provided by the XML information processor.

- In addition to information management in XML files, the system should be able to take advantage of database technology for managing XML information. A corresponding database should be automatically created based on the user-input XML schema and functions for retrieving, searching, and saving XML information from and into the database should be automatically added into the XML information processor.

- When sharing and exchanging XML information through Internet is needed, the system should be able to take advantage of Web Services technology for communicating XML information on the Web. Simple web services for updating and querying database information should be automatically provided based on the user-input XML schema and basic authentication and authorization controls should be considered for the services. In addition, functions for accessing the remote web services supported by XsEnvironment should be added into the XML information processor.

- Depending on the application scenarios of the system, different installation options should be provided for the user to selectively install XsEnvironment components. For example, installation of database and web services components is not necessary if editing of XML information is the only needed application.

## 3. SYSTEM DESIGN

According to the requirements discussed above, the major components of XsEnvironment are designed as shown in Figure 1. There are three major components: XsForm, Database, and Web Services. Their interactions are indicated by the arrowed lines. All of the three components are built automatically based on the user-input XML schema:

**a.** XsForm, which has been implemented using Microsoft Visual Basic .Net [7,8], is an XML information processor (or editor) with form-based GUI. It provides functions for editing XML documents, searching specific XML files or information in the XML files, and uploading XML files to the server. Because XsForm employs UIML [10] to describe the layout of user interfaces, it provides some flexibility for users to adjust the appearance of the tool. In this work, XsForm is being enhanced and extended to include interfaces with Database and Web Services components and corresponding information management and communication capabilities.

**b.** The database component consists of a relational database (e.g., Microsoft Access) and an accessible interface that encapsulates query and statistics functions. The approach proposed in [11] may be employed for automatically creating tables in a relational database from an XML schema.

**c.** The Web Services component is designed to support Web Services interfaces for the upload/download function of XsForm and the database component. The interface provides simple functions (or services) for the client programs to access database (including query and update operations) and upload/download XML files (or documents).

In addition, it is important to ensure that the output XML files are valid for the user-input XML Schema.

For different application purposes (or scenarios), two application modes are designed for XsEnvironment, as shown in Figure 2. For the Server Mode in Figure 2(a), the key components are Database and Web Services, while XsForm becomes an optional (as indicated by dashed lines) interface for local database operations. For the Client Mode in Figure 2(b), the database component becomes optional. Figure 3 illustrates three possible applications of Client Mode. In Figure 3(a), only XsForm is needed to edit and manage the XML files of different schemas for a stand-alone application. In this case, the upload function is not needed. Figure 3(b) shows another case for stand-alone application. A local database is used with XsForm when information management associated with large number of XML files becomes an issue. Figure 3(c) shows the client-server application case in which the upload function of XsForm is used for uploading XML files to the server. Furthermore, Figure 4 shows a more general client-server application of XsEnvironment for information sharing and exchange. In this case, one Server-Mode XsEnvironment application is interacting with several Client-Mode XsEnvironment applications on the Internet. The web services on the server respond to the clients' requests for accessing and updating the database.

## 4. CONCLUSIONS

This paper has presented the analysis and design of an XML schema-driven, tool-building environment, named XsEnvironment, for facilitating application of construction information standards expressed by XML schemas. After discussions on possible application scenarios of XsEnvironment, the system requirements have been identified and the design of the system has been proposed. Although XsEnvironment is still being implemented and its practical applications require further studies, it is believed that the automatic tool building capability based on the input XML schema, along with the proposed system design, should make XsEnvironment a convenient, user-friendly, and effective environment for dealing with editing, management, and sharing of XML documents of various schemas in the construction industry.

## REFERENCES

[1] Hsieh, S. H., H. P. Tserng, and M. T. Wang, "On Information Standardization and Sharing for Construction CALS," *Proceedings of Joint Conference of the Fifth Conference on Automation Technology and 1998 International Conference of Production Research (Asia Meeting)*, Taipei, Taiwan, R.O.C., July 20-22, 1998.

[2] Ray, E. T., *Learning XML*, Second Edition, O'Reilly & Associates, 2003.

[3] Vlist, E. V. D., *XML Schema*, O'Reilly & Associates, 2002.

[4] Gibb, B., and S. Damodaran, *ebXML: Concepts and Application*, Wiley, 2002.

[5] Segarra, S., “aecXML Domain Summary,” aecXML Technical Committee, Industry Alliance for Interoperability (http://www.iaina.org/aecxml/), 2001.

[6] Hoffman, M., “Architecture of Microsoft Office InfoPath 2003,” Microsoft Corporation, http://msdn.microsoft.com/library/en-us/odc_ip2003_ta/html/odc_inArch.asp, 2003.

[7] Hsieh, S. H., and H. T. Lin, “XsForm: a Schema-driven Form-based XML Information Processor,” *Proceedings of the 21st International Symposium on Automation and Robotics in Construction*, Jeju, Korea, September 21-25, 2004, 565-570.

[8] Lin, H. T., “Design and Implementation of an XML Schema-driven System for Form-based GUI,” M.S.

Thesis, Department of Civil Engineering, National Taiwan University, Taipei, Taiwan, 2004 (in Chinese).

[9] Newcomer, E., *Understanding Web Services: XML, WSDL, SOAP, and UDDI*, Addison-Wesley Professional, 2002.

[10] Abrams, M. and Helms, J. (Editors), "UIML 3.0 Language Specification," Harmonia Inc., USA, 2002.

[11] Wu, I. C., and S. H. Hsieh, "An UML-XML-RDB Model Mapping Solution for Facilitating Information Standardization and Sharing in Construction Industry," *Proceedings of the 19th International Symposium on Automation and Robotics in Construction*, Gaithersburg, Maryland, September 23-25, 2002, 317-321.

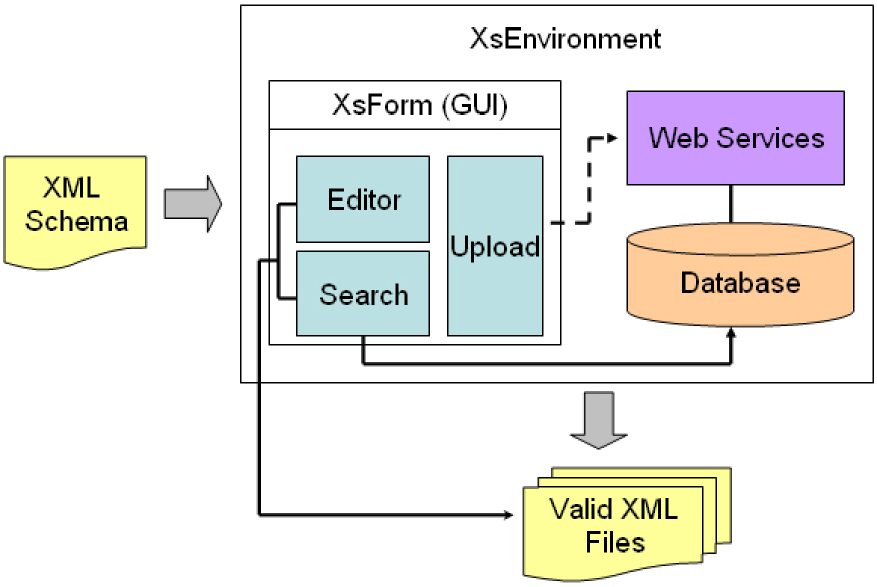

Figure 1 Major XsEnvironment components and their relationships

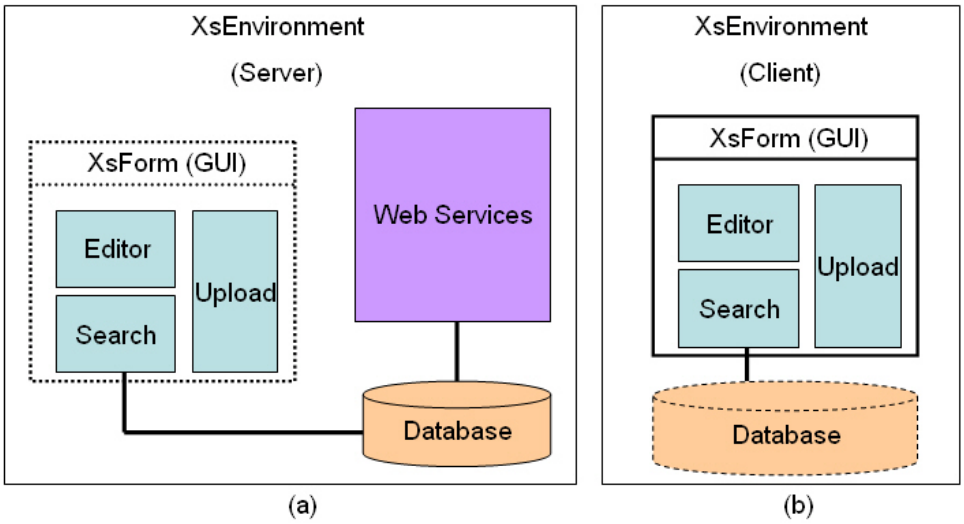

Figure 2 System architecture of XsEnvironment: (a) Server Mode and (b) Client Mode.

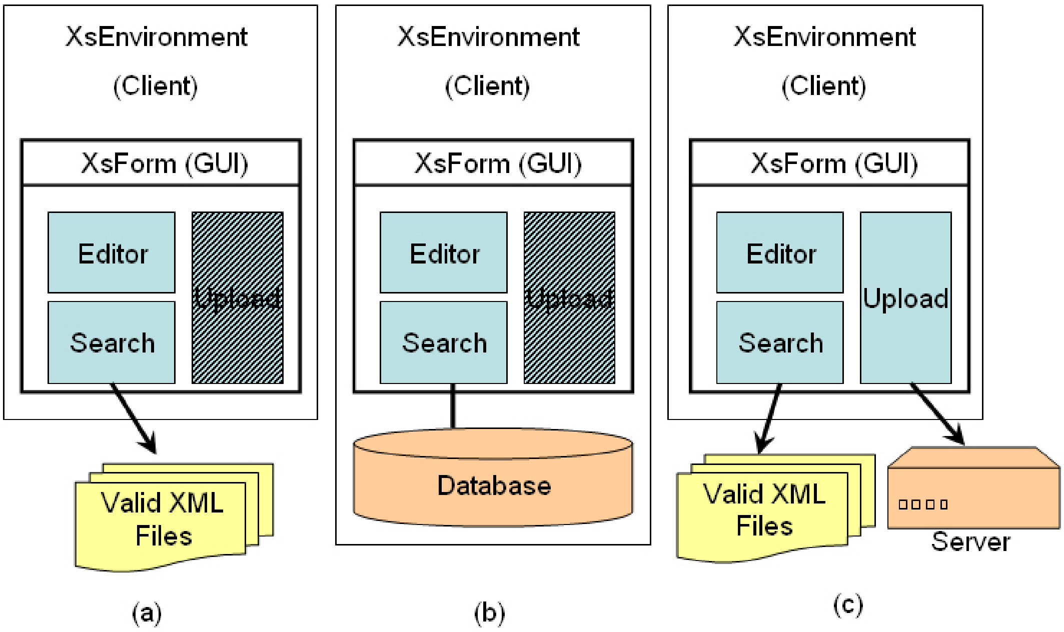

Figure 3 Three different Client Mode configurations of XsEnvironment for (a) stand-alone application without database, (b) stand-alone application with database, and (c) client-server application

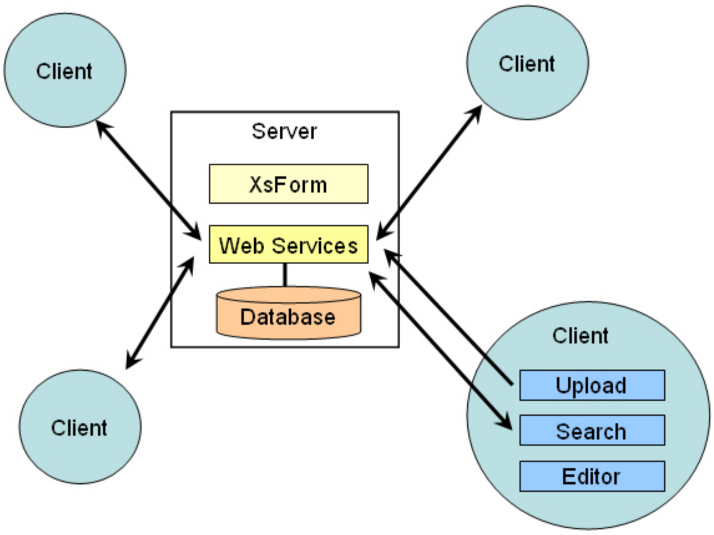

Figure 4 Client-server application of XsEnvironment using Web Services