

Design of an XML-based Document Flow Management System for Construction Projects Using Web Services

Choung-Houng Wu and Shang-Hsien Hsieh

*National Taiwan University, Department of Civil Engineering,
No.1, Sec. 4, Roosevelt Road, Taipei, Taiwan 106
boyce@caece.net, shhsieh@ce.ntu.edu.tw*

ABSTRACT: This paper presents the design of an XML-based document flow management system, called X-DFM (XML-based Document Flow Management), for increasing efficiency in processing documents in construction projects. X-DFM provides four services via Web Services: the document flow services, content management services, authorization services, and message services. The document flow services are designed based on the wFXML standard and with the consideration of flow processing security. The content management services support not only management of an XML document as a whole but also the contents in the document. The authorization services provide authorization controls of user accounts and system functions. The message services support message passing among users and between X-DFM components.

KEYWORDS: document flow management, content management, system integration, Web services, XML.

1. INTRODUCTION

Construction projects usually require sharing and processing of large amount of documents, such as reports, drawings, contracts, standards, and so on, among a number of participating parties. Furthermore, document processing flows in the lifecycle of a construction project, spanning across planning, design, construction, operation, and maintenance phases, can be quite complicated and usually involve many people of various roles in the project. Traditional paper-based document processing that relies heavily on manpower is not efficient enough to meet the need of modern construction projects that are often large and complex. In addition, the participants are often located at different places but may want to get access to the construction information or even perform document processing at their convenient time.

With rapid advancement of computer technologies in recent years, many software systems have been developed to facilitate electronic document processing for construction management. Several of them also take advantage of WWW (World Wide Web) technologies to provide Web interfaces for easy access from anywhere at anytime. However, the introduction of these advanced document-processing systems into a construction project often involves integration of

the new systems with the already existing legacy systems. From the user's point of view, all application systems involved in a project should ideally be integrated to act like a single system with uniform user interfaces, unified user account management, and integrated data management. But, from the system integration engineer's point of view, achieving this requires many difficult system integration tasks that usually involve good understanding of the underlying data structures of the legacy systems.

Because most of the legacy systems do not provide APIs (Application Programming Interfaces), one common approach to integrate with a legacy system is through direct access to its database. Although this approach may solve the integration problem at the moment, it often makes the maintenance of the integrated system difficult and causes greater complication and difficulties for future system integration tasks.

Recently, an emerging technology, called Web services [Newcomer] [Deitel], has been evolving rapidly on the World Wide Web as the need for interoperable system-to-system communication grows. Web Services technology provides a standard means of communication among different software applications and, therefore, has the potential to provide a better solution for system integration. It employs XML (eXtensible

Markup Language) [Deitel], an open standard for data exchange, to define messages for communication and how the messages are processed via common Internet protocols (e.g., HTTP). A standard reference architecture for Web services has also been proposed by the Web Services Architecture Working Group at the World Wide Web Consortium (W3C) to promote interoperability and extensibility among software applications, as well as to allow them to be integrated for more complex operations [Harold].

In this paper, the design of an XML-based document flow management system, called X-DFM, is presented. The focus of the design is on how to integrate both legacy systems and new systems within X-DFM using Web services technology so that the reusability, extensibility, and maintainability of X-DFM can be maximized. In addition, discussions are given on the design of the four services supported by X-DFM to facilitate efficient document processing in construction projects.

2. DOCUMENT FLOW MANAGEMENT

The document flow management for this work consists of two major parts. The first one is the control of the processing sequence of the document. Usually, a workflow engine is applied to achieve the management and automation of the document flow. In this research, we follow the WfMC (Workflow Management Coalition) standard to design our flow management system. The second part concerns with the review and editing of the document itself. The document may be processed and managed as a single entity or the contents of the document may need to be processed and managed as separated entities. We refer the former as document management while the later as content management. The X-DFM system designed in this work provides not only document management, but also content management.

Management System is defined by WfMC as “a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications [W3C].” Figure 1 shows the workflow reference model proposed by WfMC, which identifies components and interfaces

needed in a workflow management system. The workflow enactment service consists of one or more workflow engines to create, manage, and execute workflow instance. The interfaces around the workflow enactment service are called WAPI, which is a set of functions to access and regulate the enactment service. Process definition tools are used to define workflow processes, including the activities’ relationship, participants, criteria to start or to terminate the process, associated documents, etc. [WfMC 1999a]. Both interface 2 and interface 3 are defined for the enactment service to invoke and control other applications [WfMC 1999b]. Interface 4 is used to connect other workflow enactment services. Interface 5 is used to connect the administration and monitoring application [WfMC 1998]. The design of the workflow management services in X-DFM follows the workflow reference model shown in figure 1.

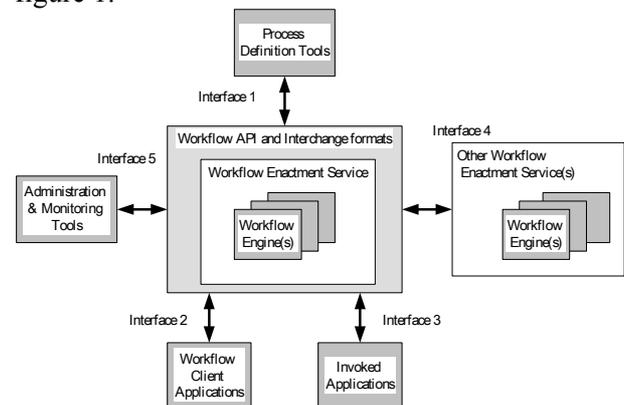


Figure 1. Workflow reference model [W3C].

The term “content management system,” as opposed to document management system, reflects the fact that such systems generally allow for breaking a document into discrete content fragments, rather than having to manage each document as a whole. Not only does this simplify such things as how to coordinate the work of multiple writers working on the same document, but also it allows users to assemble entirely new documents from existing document components. [Liu]

Generally speaking, content management systems usually provide the following capabilities:

[Chen] [Freter]

- Version control: see how a document evolves over time.
- Content sharing: see in what business processes the document content is used and re-used.

- Electronic authoring: enable users to edit and revise the contents of electronic documents.
- Electronic review: enable users to add their comments to the contents of a document without actually changing the document itself.
- Access control: define the different types of access privileges to the document contents for different types of users.
- Publishing management: control the delivery of documents to different publishing process queues.
- Workflow integration: associate the different stages of a document's life-cycle with people and projects with schedules.

The design of the content management services in X-DFM focuses on the version control, content sharing, electronic authoring, and electronic review. The authorization services and workflow management services are designed in collaboration with the content management services in X-DFM to support the services related to access control and workflow integration, respectively.

3. SYSTEM ARCHITECTURE DESIGN OF X-DFM

The architecture of the X-DFM system can be divided into three tiers as shown in figure 2: user interface, business objects, and Web Services enabled (WS-enabled) application servers, which consist of both newly-developed systems and legacy systems with Web services adapter. Four services are provided by X-DFM through WS-enabled application servers: the document flow services, content management services, authorization services, and message services. A brief description on the major components in the architecture shown in figure 2 is given as follows:

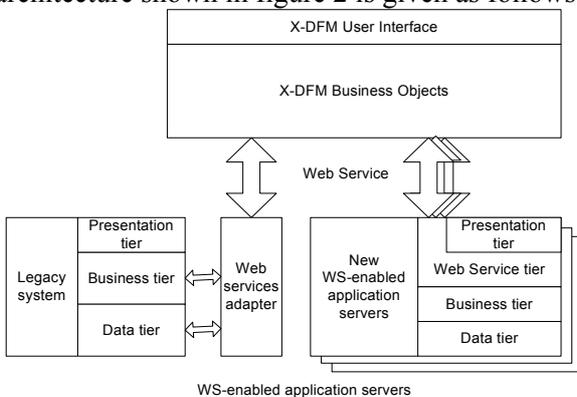


Figure 2. Design of the X-DFM architecture.

- X-DFM user interface: It is designed to provide personalized web-based GUI (Graphic User Interface) for easy access and ease-of-use. With an appropriate account, the user can login X-DFM through Internet or Intranet from any places and at any time using a common web browser, e.g., Internet Explorer. Upon login, the user will see a list of messages containing notification and instructions of all document-processing jobs he or she is expected to perform at the moment. Figure 3 shows the document-processing flow for a user. The user first retrieves one of the messages in the list from the message services. Based on the instruction in the message, the user then tries to check out the documents needed for performing the job from the content management services. After checking the user's authorization privilege by authorization services, X-DFM returns to the user only the contents in the documents he or she is asked and allowed to review and/or modified. When the user completes the document-processing job and checks in the documents, he or she then notifies the document flow services that will then follow the pre-defined flow for next-stage document-processing.

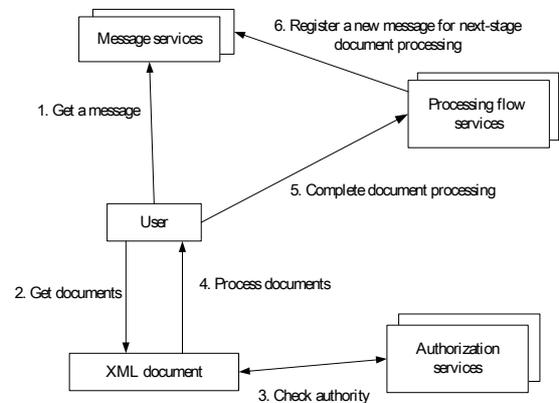


Figure 3. System interaction diagram of X-DFM.

- X-DFM business objects: X-DFM business objects are classified into two types of components: interface components and business components. The interface components are designed to support the implementation of the X-DFM GUI, while the business components provide internal business transaction results and information integration from different WS-enabled application servers.
- WS-enabled application servers: all of the services provided by X-DFM are through a set

of WS-enabled application servers that are integrated by the X-DFM business objects. The WS-enabled application servers can be newly developed systems that support Web services or legacy systems with Web services adapter. Because Web services technology is employed to support loosely coupled integration between the business objects and the WS-enabled application servers, adding new services in X-DFM can be done easily by adding the corresponding new WS-enabled application server and business objects without the need of modifying existing WS-enabled application servers. Design of WS-enabled application servers for X-DFM will be discussed further in Section 4.

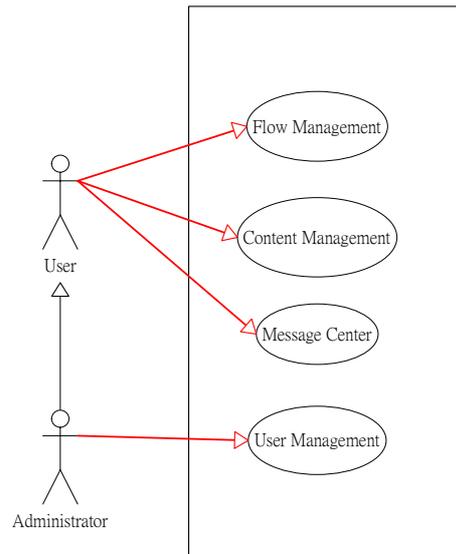


Figure 4. The X-DFM system use case diagram.

Web services technology is employed in the architecture design of X-DFM (see figure 2) to achieve flexibility in system development, to ease the system integration task, and to increase system extensibility, maintainability, and reusability. Services in X-DFM can be developed independently in the form of WS-enabled application servers that may run on machines with different operating systems and distributed at different locations. In addition to the newly developed WS-enabled application servers, legacy systems can be made WS-enabled through the design of Web services adapters and integrated into X-DFM. With standard communication protocols established between X-DFM business objects and WS-enabled application servers using Web services, the addition of new application servers and replacement of old application servers by other equivalent application servers in X-DFM can be done easily with minimized system modification. Furthermore, because the WS-enabled application servers are independent servers, they provide services to not only X-DFM but also other systems through Web services.

4. DESIGN OF WS-ENABLED APPLICATION SERVERS

X-DFM provides four major services to help engineers in concurrent document processing: authorization services, the document flow services, content management services, and message services. There are four major corresponding subsystems in X-DFM to support the four services: user management system, flow management system, content management system, and message center (see figure 4):

- User management system: It allows the administrator to manage user accounts, organize user groups, and perform authorization tasks.
- Flow management system: Its design follows the WfMC standard and the Wf-XML specification. It allows users to pre-define commonly used process flows and save them as templates. The user can then use the pre-defined process flow from templates or create a new process flow for the task at hand.
- Content management system: It provides services for version control, content sharing, electronic authoring, and electronic review of the document contents.
- Message center: Two types of messages are handled by the message center. One type is a notice from the flow management system to remind the users about their document-processing tasks. Another type is a correspondence between the users. If one leaves a message to another in the message center. The message is then delivered thru the e-mail system or posted on the recipient's X-DFM working webpage.

Every subsystem in X-DFM is designed as an independent WS-enabled application server. It usually has its own user interface and provides services to X-DFM or other clients via Web services. For a newly developed WS-enabled application server, a 4-tiers architecture framework is proposed (see figure 2). Different from the common 3-tiers architecture, consisting of presentation tier, business logic tier, and data

tier, a Web service tier is added between the presentation tier and the business logic tier in this 4-tiers architecture. The Web service tier is designed to enable Web services on all business functions of the system. It also provides a layer of encapsulation for all business objects in the business logic tier. The graphical user interfaces in the presentation tier are implemented using the business functions (in the form of Web services) provided by the Web service tier. The clients can obtain services from the system through either user interfaces provided in the presentation tier or direct interactions with the services in the Web service tier.

For taking advantage of legacy systems in this framework, a Web services adapter is needed to turn a legacy system into a WS-enabled application server. The implementation of the Web service adapter may involve interactions with the business tier and the data tier of the legacy system. If the legacy system supports an API (Application Programming Interface) or allows direct access to the business objects in the business tier, the Web service adapter can be implemented using the API and/or the business objects. Otherwise, the Web service adapter is implemented through direct interactions with the database in the data tier of the legacy system.

5. COMPONENTS DESIGN IN X-DFM

In this section, the authorization service component is used as an example to discuss the design of components in X-DFM.

Figure 5 shows the design of the authorization service component in X-DFM. The ACL (Authorization Control List) class between the CIdentity and CAnyObject classes is designed to establish authorization relationships between the system objects that need access control (e.g., files, documents, and operations) and the users. Users may have various authorization privileges on system objects at the same time depending on their job functions and the roles they play in a project. Both the CIdentity and CAnyObject classes are abstract super classes and have no instances. All objects of the derived classes of the CAnyObject class have their access controls managed by a set of ACL objects.

From the point of view of authorization, the ACL component is in the Web service tier, while the rest of the components is in the business tier of the

WS-enabled authorization service application server (see figure 5). In X-DFM, to set up the access privilege of a system object for a user, the authorization object in the business tier of X-DFM sends a request to the ACL's Web service with a simple XML message like this:

```
<AUTHORIZATION system="X-DFM"
method="set">
  <IDENTITY>boyce</IDENTITY>
  <OBJECT>000001</OBJECT>
  <PERMISSION>GRANTED</PERMISSION>
</AUTHORIZATION>
```

In the message, the IDENTITY value can be the value of ISID of the CUser class or the CRole class, both inheriting from the CIdentity class. The OBJECT value can be the value of OID of the classes inheriting from the CAnyObject class. The PERMISSION value can be set GRANTED or DENIED. If it is not set, the default value is NODATA.

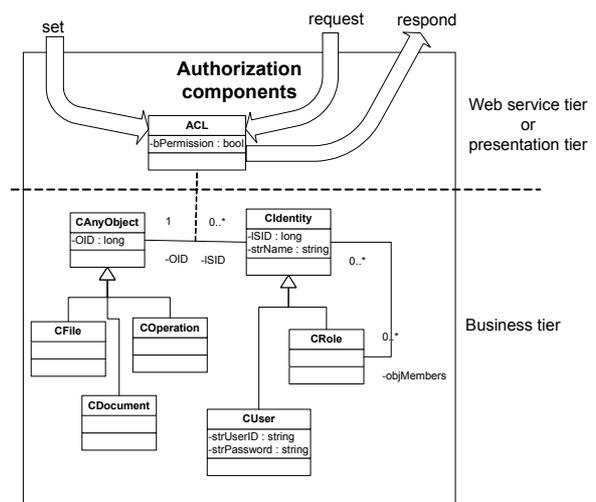


Figure 5. Design of authorization service components in X-DFM.

For checking the access privilege of a system object for a user, a simple request message in XML sent to the ACL's Web service is like this:

```
<AUTHORIZATION system="X-DFM"
method="request">
  <IDENTITY>boyce</IDENTITY>
  <OBJECT >000001</OBJECT>
</AUTHORIZATION>
```

The XML message responded from the ACL service is then like this:

```
<AUTHORIZATION system="X-DFM"
method="respond">
  <IDENTITY >boyce</IDENTITY>
  <OBJECT>000001</OBJECT>
  <PERMISSION>GRANTED</PERMISSION>
</AUTHORIZATION>
```

6. CONCLUSIONS

In this paper, the design of an XML-based document flow management system, called X-DFM (XML-based Document Flow Management), has been presented for facilitating document-processing tasks in a construction project. The design takes advantage of the evolving Web services technologies to achieve flexibility in system development, to ease the system integration task, and to increase system extensibility, maintainability, and reusability. The difficult issue on integration of new application systems and legacy systems has been addressed in the design of X-DFM. In addition, four essential services have been proposed and designed in X-DFM to support document flow management services: the document flow services, content management services, authorization services, and message services. It is hoped that the design ideas presented in this paper can shed some light on designing construction management systems with great flexibility, extensibility, maintainability, and reusability.

REFERENCES

[Newcomer] Newcomer, E. 2002, *Understanding Web Services*, Addison-Wesley, Boston, USA.

[Deitel] Deitel, H. M., P. J. Deitel, B. DuWaldt, and L. K. Trees, 2003, *Web Services - A Technical Introduction*, Prentice Hall, New Jersey, USA.

[Harold] Harold, E. R., and W. S. Means, 2001, *XML in a Nutshell: A Desktop Quick Reference*, O'Reilly & Associates, Sebastapol, CA, USA.

[W3C] World Wide Web Consortium, 2002, "Web Services Architecture Requirements," W3C Working Draft, November 2002 (available at <http://www.w3.org/TR/wsa-reqs>).

[WfMC 1999a] Workflow Management Coalition, 1999. "Terminology & Glossary," Issue 3.0, The Workflow Management Coalition Specification, February 1999

[WfMC 1999b] Workflow Management Coalition, 1999. "Interface 1: Process Definition Interchange Process Model," version 1.1, The Workflow Management Coalition Specification, October 1999.

[WfMC 1998] Workflow Management Coalition, 1998. "Workflow Management Application Programming Interface (Interface 2&3) Specification," version 2.0, The Workflow Management Coalition Specification, July 1998.

[Liu] Liu T.-H., 2000. "Design and Implementation of a CITIS-Based Information Sharing System," M.S. Thesis, Department of Civil Engineering, National Taiwan University.

[Chen] Chen C.-Y., 2001. "Design and Implementation of XML-based Content Management System Using RDB," M.S. Thesis, Department of Civil Engineering, National Taiwan University.

[Freter] Freter, T., 2000. "XML: Document and Information Management," <http://www.sun.com/980908/xml/>