E-Transcript Web Services System Supporting Dynamic Conversion Between XML and EDI

Myungjae Kwak
Claremont Graduate University, myungjae.kwak@cgu.edu

Woohyun Kang
Claremont Graduate University, woohung.kang@cgu.edu

Gondy Leroy
Claremont Graduate University, gondy.leroy@cgu.edu

Samir Chatterjee
Claremont Graduate University, samir.chatterjee@cgu.edu

E-Transcript Web Services System Supporting Dynamic Conversion Between XML and EDI

Myungjae Kwak
School of Information Systems and Technology
Claremont Graduate University
Myungjae.Kwak@cgu.edu

Woohyun Kang
School of Information Systems and Technology
Claremont Graduate University
Woohung.Kang@cgu.edu

Gondy Leroy
School of Information Systems and Technology
Claremont Graduate University
Gondy.Leroy@cgu.edu

Samir Chatterjee
School of Information Systems and Technology
Claremont Graduate University
Samir.Chatterjee@cgu.edu

ABSTRACT

As XML becomes a standard for communications between distributed heterogeneous machines, many schools plan to implement Web Services systems using the XML e-transcript (electronic transcript) standard. We propose a framework that supports both XML e-transcript Web Services and existing EDI e-transcript systems. The framework uses the workflow engine to exploit the benefits of workflow management mechanisms. The workflow engine manages the e-transcript business process by enacting and completing the tasks and sub-processes within the main business process. We implemented the proposed framework by using various open source projects including Java, Eclipse, and Apache Software Foundation's Web Services projects. Compared with traditional EDI systems, our proposed system was evaluated to have higher reusability and scalability and save developers time and cost. It also provides users with rich look and feel user interfaces.

Keywords

Electronic Transcript, EDI, Web Services, Workflow Management System, XML

INTRODUCTION

Due to its structured data model, eXtensible Markup Language (XML) has been a de facto standard for data communication between machines in many areas, such as e-commerce and electronic medical records. Recently, there have also been efforts to use XML to process e-transcripts that contain academic records of students in educational institutions (www.pesc.org). There are several thousand schools and educational institutions in the US, and millions of students are transferring to other schools or applying to higher educational institutions every year. Previously, many schools have used Electronic Data Interchange (EDI) systems and a central FTP server to transmit or process e-transcripts. Currently, however, most schools are moving towards XML e-transcripts because of their increased machine and human readability and interoperability.

Accordingly, many schools plan to develop a flexible and scalable e-transcript processing system, such as Web Services systems, to process XML e-transcripts. However, many are still using EDI systems and a FTP server as a brokering server. Therefore, the newly developed systems need to support both XML and EDI e-transcripts and to allow both Web Services and FTP transmission mechanisms.

In this study, we propose a framework to support both XML and EDI e-transcript and both Web Services and FTP transmission mechanisms by referring to a registry such as Universal Description Discovery and Integration (UDDI). The proposed framework is implemented based on the workflow reference model suggested by the Workflow Management Coalition (WfMC) to help developers implement workflow management systems for managing business processes.

This paper consists of the following sections. Section 2 describes Web Services, e-transcript standards efforts, and workflow management systems. Section 3 analyzes the requirements for a framework that supports Web Services. Section 4 discusses our framework and explains the core models of our framework. Section 5 describes implementation environments and user interfaces. Section 6 evaluates the proposed framework. As a conclusion to the paper, we summarize our study.

RELATED WORKS

Electronic Document Interchange (EDI) and Web Services

EDI has been proposed as a data transmission medium between firms including agents or intermediaries and many companies adopted it to communicate electronically each other for various purposes (Hill and Ferguson, 1989). The emergence of EDI has solved the problem (e.g., labor and time cost) of paper-based systems by sending structured, paperless messages from one agent to another agent. However, as network economy develops rapidly, companies must be able to respond quickly to changing environments by adopting effective and efficient information technology. In this light, EDI achieves electronic data interchange, but it is a complex technology, which does not provide the required flexibility (Khoumbati, Themistocleous, and Irani, 2005). Especially since many companies and agents develop heterogeneous information systems for their own purposes using different technologies. Since late 1990, XML and Web Services have been developed to enable these heterogeneous applications written in diverse languages to interoperate more easily and less expensively than other traditional methods such as EDI (Khoumbati, et al., 2005).

Since the Web Services architecture (Gottschalk, Graham, Kreger, and Snell, 2002) was proposed by the World Wide Web Consortium (W3C) in early 2000, it has been used in numerous fields with various purposes. Web Services have been considered the best architecture for requesting services and providing services using Repository or UDDI. Web Services have been used to implement the interoperation between machines over a network. For implementation of the Web Services, W3C proposed core specifications such as XML (Goldfarb and Prescod, 1999), SOAP (Ferris and Farrell, 2003; Curbera, Duftler, Khalaf, Nagy, Mukhi, and Weerawarana, 2002), and WSDL (Chinnici, Gudgin, Moreau, Schlimmer, Weerawarana, 2004). Besides, the Organization for the Advancement of Structured Information Standards (OASIS) also proposed a strong component, such as UDDI (Curbera, et al., 2002), to provide a repository as a broker for both a service provider and a requester.

In the last few years, several researchers have studied the design framework of Web Services and proposed some new frameworks and design models (Gottschalk, et al., 2002; Kreger, 2001; Sollazzo, Handschuh, Staab, Frank, 2002). Most of these adopted standard architecture and provided broad approaches and examples to build up web services systems for their academic purposes. Their results were often insufficient for understanding Web Services frameworks in real world. With respect to representing real frameworks and examples, our system shows a form of real world Web Services systems, because this system was physically developed based on actual requirements of a state university that processes many electronic transcripts.

Electronic Transcript Standards

The Post-secondary Electronic Standards Council (PESC) has proposed and developed an EDI standard for e-transcripts. Accordingly, many universities have implemented the EDI System using an FTP server (www.pesc.org). The most prominent implementation of PESC EDI standard is SPEEDE (Standardization of Postsecondary Education Electronic Data Exchange) system of University of Texas, Austin (http://registrar.utexas.edu/speede/). This SPEEDE system currently uses ANSI X-12 EDI transaction set and a SPEEDE FTP server as a brokerage facility, which relays EDI transcripts from many academic institutions to their destinations. As XML becomes a de facto standard for communication between information systems, PESC has issued several XML e-transcript standards to help developers implement e-transcript systems since 2005 (www.pesc.org). Accordingly, the SPEEDE system announced the plan to accept XML standards but did not pronounce how to admit and process XML transcripts or how to convert them to EDI transcripts. Also it did not publish whether or not it will adopt the Web Services architecture. None of the related contributions examined acknowledges the framework that combines the dynamic conversion of XML/EDI transcripts and the integration of Web Services system with EDI systems. Therefore, our proposed framework may be the first attempt to adopt XML/EDI dynamic conversion module based on the Web Services architecture for transacting electronic transcripts.

Workflow Management System

The Workflow Management Coalition (WfMC) defined the mechanisms for workflow management systems as follows: to decide, initiate, and enact business tasks and processes (Kwak, Shim, and Han, 2002). Based on those defined mechanisms, various Workflow Management Systems (WfMSs) have been suggested to support the automation of business processes in a distributed heterogeneous environment (Hayes et al., 2000; Kim et al., 2000; Dogac et al., 1998) and to provide a basis for integration and interoperability capability between different products. In those WfMSs, a workflow can be defined as "a specific representation of a process, which is designed in such a way that formal coordination mechanisms between activities, applications, and process participants can be controlled by an information system, the so-called workflow management system (Muehlen, 2004)." By using this workflow management mechanism, our proposed system can initiate the business processes, monitor their accurate states, and provides more reliable mechanism than other EDI systems.

REQUIREMENTS FOR FRAMEWORKS SUPPORTING WEB SERVICES FOR E-TRANSCRIPTS

Many schools have already adopted EDI e-Transcript standards (www.pesc.org) and have used a FTP server (http://registrar.utexas.edu/speede/) as a brokering repository as shown in Figure 1. Recently, schools have started to use XML e-transcript standards proposed by PESC, and most of them are using both EDI and XML e-transcript standards. This is because the many existing modules for e-transcripts were implemented for EDI standards and it is costly to completely replace them with XML standards. Moreover, most schools are using various student information systems from different vendors. In this light, schools that use various heterogeneous systems require a flexible framework that allows various heterogeneous systems and also is able to accept both XML and EDI standards.

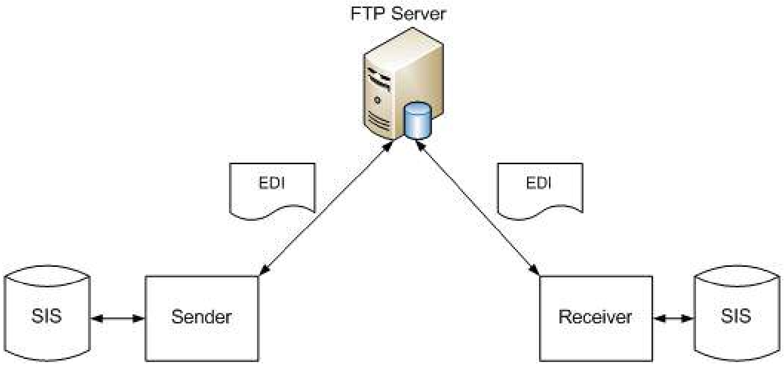


Figure 1. Existing e-Transcript EDI system

On the other hand, sending e-transcripts is a business process, which means it may contain various sub-processes or tasks. The e-transcript business process inevitably involves other schools' systems and may involve other systems within the school, such as student information systems. Moreover, during the execution of business processes, the status of the business processes and various exceptions should be monitored and handled to preserve their integrity. Previous research and implementations show that the status and exceptions related to the execution of business processes can be managed by workflow management systems.

A FRAMEWORK FOR E-TRANSCRIPTS WEB SERVICES SYSTEM

Design Principle

Service requesters and providers should not need to think too much about the form of e-transcript standards and the type of transmission mechanisms. The conversion of XML to EDI transcripts or EDI to XML transcripts and the transmission of the e-transcript through a proper mechanism (either Web Services or FTP server) should be performed at run time. Web Services systems, combined with workflow management system, can support dynamic conversion between EDI and XML and dynamic switching between Web Services and FTP server. The framework can be achieved using a sub-process model, a dynamic state transition model, and an exception handling model. These models are designed and implemented using object-oriented concepts and software design patterns, such as dynamic binding, encapsulation, delegation pattern, and proxy pattern (Kwak, et al., 2002).

Sub-process Model

WfMC defined a workflow process as a mechanism to automate business processes according to a defined set of rules to achieve an overall business goal (Dogac et al., 1998; Object Management Group, 2000). As shown in Figure 2, a task can stand for alternative sub-processes (Kwak, et al., 2002). We can anticipate many advantages from this model; we can manage much simpler main processes, and the sub-processes can be mapped to the main processes, not at build time but at run time (Kwak, et al., 2002; Schmidt, 1999).

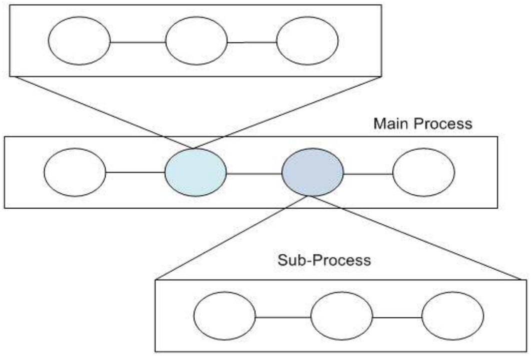


Figure 2. Example of Sub-process Model

Dynamic State Change Model

We can consider the workflow engine a state transition model, by which internal or external events, or specific control decisions by a workflow engine, change the states of individual processes (Object Management Group, 2000). The proposed framework supporting Web Services and workflow management is based on the dynamic sub-process model, where decisions for selecting proper sub-processes and tasks are created at run time. That is, sub-processes can be dynamically mapped to a main process. The status change model should be able to express the states of the sub-processes made at run time to provide users correct monitoring information (Kwak, et al., 2002).

In order to monitor correctly the status of the process, we use a dynamic status transition model as defined in Figure 3. This transition model affects all processes' and tasks' instances, and the workflow engine manages the states of the sub-processes' and tasks' instances using sub-process manager.

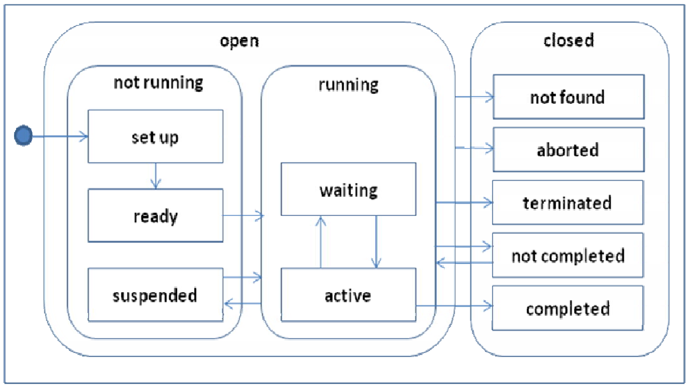


Figure 3. Dynamic State Transition Model

Exception Handling Model

When systems send and receive electronic transcripts, various exceptions can occur. Those exceptions should be handled to make the business process reliable, for example, UDDI may not have any information for the receiving institutions. Or receiving institutions' systems may encounter difficulties during the execution of their business processes. To handle these exceptions and provide administrators with correct status information about the business process, we defined four possible states other than the completed state (Kwak, et al., 2002). They are shown in Figure 3.

(1) **Not Found**. In this situation, the initiating system cannot find the information for the target institution. An administrator's intervention is required to solve this situation.
(2) **Aborted**. In this situation, the target institution's system cannot perform the task because the service interface was changed. An administrator's intervention is also required to solve this situation.
(3) **Terminated**. In this situation, the target institution's sub-process has started but some of the tasks cannot be completed because of some exceptions.
(4) **Not completed**. In this situation, the target system has started the sub-process but the target system aborted the sub-process and restarted an alternative sub-process.

System Architecture

Our proposed system supports the runtime conversion between e-transcript standards and the runtime switching between Web Service and FTP in distributed heterogeneous and Internet environments. Figure 4 describes the overall architecture of the proposed framework. In our framework, three features, such as workflow engine, XML-To-EDI conversion module, and security module, serve critical roles.

In the framework, schools that want online e-transcript transactions with other schools can publish, update, and delete their systems' information by sending WSDL to the UDDI (Kreger, 2001). When a student comes to an administrator, or accesses a school web site, the business process to send e-transcripts can be started by the administrator or enacted by the user through the workflow engine. Before the workflow engine starts the e-transcript sending sub-process in the middle of the main business process, the Transcript Decision Manager of the workflow engine would determine the proper standards (either an XML or EDI standards). It would also determine the proper communication mechanism (either Web Services or FTP server) by enacting the e-transcript sending sub-process, which refers to UDDI and decides a proper standard and proper transmission mechanism.

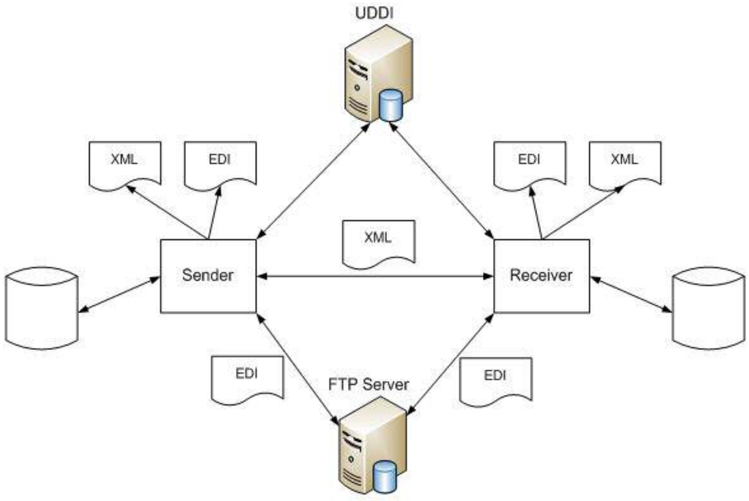


Figure 4. Overall Architecture of Proposed Web Services Framework

Workflow Engine

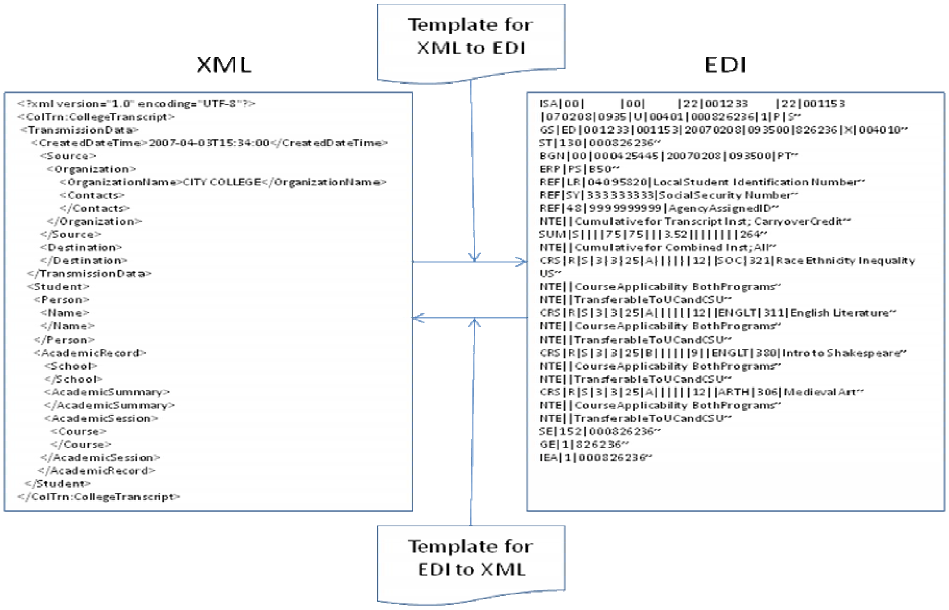
The workflow engine needs a flexible architecture to select and bind sub-processes at runtime based on the information about the systems of target schools. Once the workflow engine finds the proper standard and mechanism, it creates a Transcript Handler, workflow manager object that manages an external sub-process, and delegates the responsibility to the handler to start the e-transcript sending sub-process (Kwak, et al., 2002). The Transcript Handler creates either a XML or an EDI e-transcript and sends the e-transcript based on the selected medium. In this light, the workflow management engine provides users with implementation transparency for the specific sub-processes since the administrator can send e-transcripts without knowledge of the types of e-transcript standards and transmission mechanisms (Kwak, et al., 2002).

Dynamic XML to EDI Conversion Module

To convert XML e-transcripts to EDI e-transcript, or EDI to XML, we designed a XML-to-EDI Conversion Manager and two templates: XML-to-EDI template and EDI-to-XML template. Based on the information for the target school, the XML-to-EDI Conversion Manager uses either of the two templates at run time to create proper e-transcripts.

Security Module

E-transcripts usually contain confidential data, such as social security number and course grades. Therefore, to prevent forgery, it is important to secure the e-transcript while it is being transmitted between schools. To secure data integrity, it is required to zlib compress and base64 encode e-transcripts according to the recommendations from PESC (www.pesc.org). Digitally signing the e-transcripts with X.509 certificates is also a necessity to prevent forgery. When e-transcripts are transmitted, the sender signs the hashed message digest with a private key, and the receiver verifies the e-transcripts with the sender's public key. On the receiver side, the *PublicKeyHandler* is created to obtain the sender's public key and verify the integrity of the received e-transcripts (www.pesc.org).



```
<?xml version="1.0" encoding="UTF-8"?>
<ColTrn:CollegeTranscript>
 <TransmissionData>
   <CreatedDateTime>2007-04-03T15:34:00</CreatedDateTime>
     <Source>
      <Organization>
        <OrganizationName>CITY COLLEGE</OrganizationName>
        <Contacts>
        </Contacts>
      </Organization>
     </Source>
     <Destination>
     </Destination>
  </TransmissionData>
  <Student>
   <Person>
    <Name>
    </Name>
   </Person>
   <AcademicRecord>
     <School>
     </School>
     <AcademicSummary>
     </AcademicSummary>
     <AcademicSession>
      <Course>
      </Course>
    </AcademicSession>
   </AcademicRecord>
 </Student>
</ColTrn:CollegeTranscript>
```

```
ISA|00|          |00|          |22|001233        |22|001153
|070208|0935|U|00401|000826236|1|P|S~
GS|ED|001233|001153|20070208|093500|826236|X|004010~
ST|130|000826236~
BGN|00|000425445|20070208|093500|PT~
ERP|PS|B50~
REF|LR|04095820|LocalStudent Identification Number~
REF|SY|333333333|SocialSecurity Number~
REF|48|9999999999|AgencyAssignedID~
NTE||Cumulativefor Transcript Inst; CarryoverCredit~
SUM|S||||75|75|||3.52||||||||264~
NTE||Cumulativefor Combined Inst;All~
CRS|R|S|3|3|25|A||||||12||SOC|321|Race Ethnicity Inequality
US~
NTE||CourseApplicability BothPrograms~
NTE||TransferableToUCandCSU~
CRS|R|S|3|3|25|A||||||12||ENGLT|311|English Literature~
NTE||CourseApplicability BothPrograms~
NTE||TransferableToUCandCSU~
CRS|R|S|3|3|25|B||||||9||ENGLT|380|Intro to Shakespeare~
NTE||CourseApplicability BothPrograms~
NTE||TransferableToUCandCSU~
CRS|R|S|3|3|25|A||||||12||ARTH|306|Medieval Art~
NTE||CourseApplicability BothPrograms~
NTE||TransferableToUCandCSU~
SE|152|000826236~
GE|1|826236~
IEA|1|000826236~
```

Figure 4. XML to EDI Conversion Module

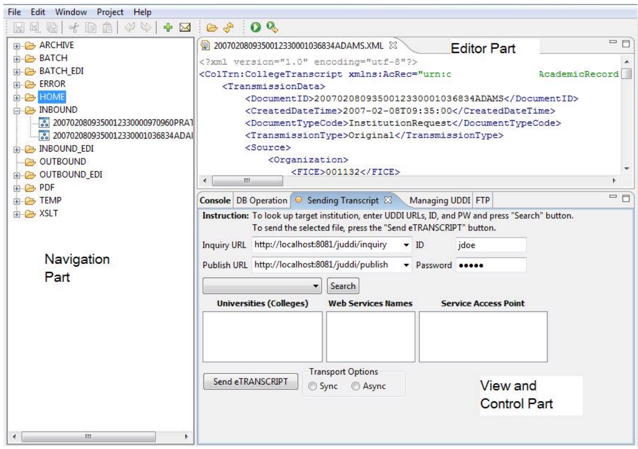
IMPLEMENTATION

Development Environment

The proposed framework has been implemented using various open source projects. We used Java 1.5 and Eclipse 3.3 Rich Client Platform as programming language and development tools. We also used Spring Frameworks, Apache Axis 2.0 and jUDDI as Web Services and repository APIs (Application Programming Interfaces).

Interface Design

To provide rich client application to the administrator, we designed and created user interfaces with Rich Client Platform by Eclipse. The rich client platform provides an interface with a considerably rich look and feel for users, as shown in Figure 5.



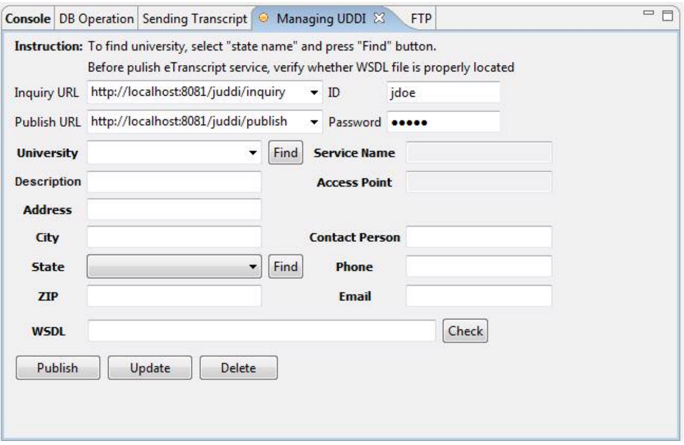
2007020809350012330001036834ADAMS.XML

```
<?xml version="1.0" encoding="utf-8"?>
<ColTrn:CollegeTranscript xmlns:AcRec="urn:c                    AcademicRecord
    <TransmissionData>
        <DocumentID>2007020809350012330001036834ADAMS</DocumentID>
        <CreatedDateTime>2007-02-08T09:35:00</CreatedDateTime>
        <DocumentTypeCode>InstitutionRequest</DocumentTypeCode>
        <TransmissionType>Original</TransmissionType>
        <Source>
            <Organization>
                <FICE>001132</FICE>
```

Instruction: To look up target institution, enter UDDI URLs, ID, and PW and press "Search" button.
To send the selected file, press the "Send eTRANSCRIPT" button.

Inquiry URL http://localhost:8081/juddi/inquiry ID jdoe

Publish URL http://localhost:8081/juddi/publish Password •••••

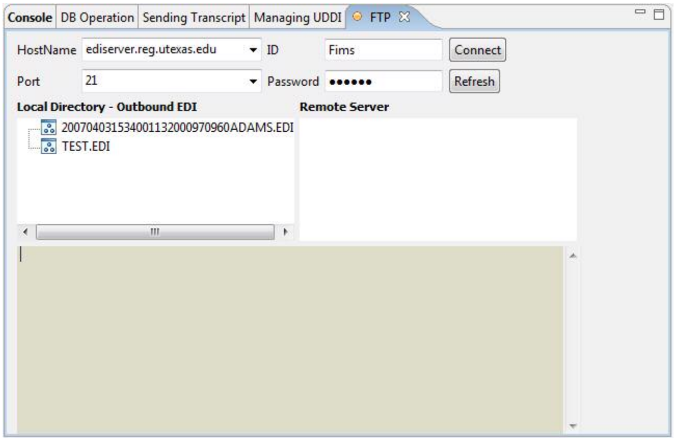


Instruction: To find university, select "state name" and press "Find" button.
Before pulish eTranscript service, verify whether WSDL file is properly located

Inquiry URL http://localhost:8081/juddi/inquiry ID jdoe

Publish URL http://localhost:8081/juddi/publish Password •••••

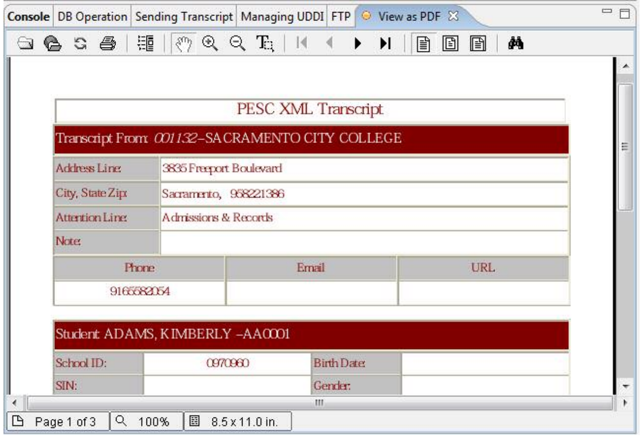
Figure 5. Main User Interface **Figure 6. Interface for Managing UDDI**



HostName: ediserver.reg.utexas.edu
ID: Fims
Port: 21
Password: ••••••

Local Directory - Outbound EDI
- 20070403153400113200097096OADAMS.EDI
- TEST.EDI

Remote Server



PESC XML Transcript

Transcript From: *001132*–SACRAMENTO CITY COLLEGE

Address Line:	3835 Freeport Boulevard
City, State Zip:	Sacramento, 958221386
Attention Line:	Admissions & Records
Note:	

Phone	Email	URL
9165582054		

Student: ADAMS, KIMBERLY –AA0001

School ID:	0970960	Birth Date:	
SIN:		Gender:	

Page 1 of 3 | 100% | 8.5 x 11.0 in.

Figure 7. Interface to Send E-Transcript Using FTP Server **Figure 8. Interface to View E-Transcript as PDF File**

The main user interface provides three main parts: document navigator, editor, and control. The document navigator is used to search and manage e-transcripts as files. The editor is used to modify the electronic transcript documents. The control is used when we send XML transcripts based on Web Services (Figure 5) and store the received e-transcripts into databases of student information systems. It is also used to publish or update service information to UDDI (Figure 6), send the EDI transcript using FTP server (Figure 7), and view an e-transcript as a PDF file using XSLT (Figure 8).

EVALUATION

In order to evaluate the proposed system, we adopted four major stakeholder's perspectives (i.e., discovery, semantics, security, and reliability) from the W3C Working Paper (W3C, 2004) about Web Services architecture. We also derived four measurement criteria from previous research (Hix et al., 2004) for assessing the improvements of proposed systems. To evaluate those eight categories, we designed ten questions to obtain evaluations from five system experts, consisting of two Web Services researchers and three system developers. In the evaluation session, we explained both traditional EDI systems and our proposed system by using detailed descriptions of both systems. We also demonstrated both systems before asking those ten questions.

We used a five-point Likert scale to measure experts' responses for each question. Table 1 shows a summary of the experts' evaluations of both traditional EDI systems and the proposed system. The result shows that our system is better than the traditional EDI systems in terms of the eight perspectives.

Systems Measures	Existing Traditional Systems (EDI)	Proposed System (Web Services)
Core Requirements		
Discovery	*Low*	*Medium-High*
Semantics	*N/A*	*Medium*
Security	*Medium*	*Medium-High*
Reliability	*Medium*	*Medium-High*
Improvements		
Reusability	*Low*	*High*
User Interface	*Low*	*High*
Time Saving	*Low*	*Medium*
Cost Efficiency	*Medium*	*Medium-High*

Table 1. The Summary of Expert Evaluation

Compared to the traditional EDI e-transcript systems, the discovery and semantics aspects of our proposed system are much more flexible and scalable than traditional systems that only use EDI transcripts through a FTP server. Our proposed system can support both XML and EDI e-transcript standards and allow both Web Services transmission and FTP transmission mechanisms. From a security perspective, our system provides a security-handling module based on the Public Key Infrastructure and a Digital Certificate to secure the electronic transcript. Traditional systems use a general HTTPS protocol to transact electronic transcripts. From the reliability perspective, we adopted a dynamic state transition and exception handling model to allow administrators to monitor the status of business processes and take care of exceptions situations. The experts determined that it increases the reliability of the proposed system. However, traditional EDI systems do not allow the sender's system to directly monitor the status of the business process, only notify the status of the business process by sending e-mail messages to administrator. This was evaluated as being less reliable because the e-mail messages can be lost or misdirected.

From the reusability perspective, our proposed system was evaluated to have high reusability in that it easily allows new Web Services systems of other educational institutions to join this framework by simply publishing their information to UDDI. Moreover, adopting Web Services architecture can reduce the maintenance complexity and require less programming skill (Khoumbati, et al., 2005). These characteristics increase the reusability of the proposed system. From the user interface perspective, our proposed system adopted the Rich Client Platform and was evaluated to provide users with a high level of look and feel interfaces. However, traditional EDI systems do not have a unified user interface and users have to use an FTP client application, which has more complicated interfaces for general users. From the time and cost perspective, we used various open source projects, which dramatically saved us development time and cost. But EDI systems usually require high initial capital investments and their implementation is not simple (Khoumbati, et al., 2005). Accordingly, it may take more time to develop and deploy EDI systems.

CONCLUSION

In this paper, we proposed an electronic transcript Web Services framework supporting not only XML and EDI e-transcripts, but also both Web Services and FTP transmission mechanisms. We analyzed the requirements of electronic transcript systems and defined the design principles and three core models. By using various open source projects, we implemented the proposed framework based on Web Services architecture and workflow management mechanism.

We had five experts evaluate our proposed system in terms of eight perspectives, which were derived from the Web Services architecture recommendation of W3C and previous research. The proposed system was evaluated to be much better than traditional EDI systems in terms of those eight perspectives.

REFERENCES

1. Chinnici, R., Gudgin, M., Moreau, J. J., Schlimmer, J., & Weerawarana, S. (2004) Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C Working Draft, 26.
2. Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., and Weerawarana, S. (2002) Unraveling the Web services web: an introduction to SOAP, WSDL, and UDDI, *Internet Computing, IEEE*, 6, 2, 86-93.
3. Dogac, A., Kalinichenko, L., Ozsu, M. T., and Sheth, A. (ed) (1998) Workflow Management Systems and Interoperability, NATO ASI Series, Springer-Verlag, Berlin.
4. Ferris, C., and Farrell, J. (2003) What are Web Services? *Communications of the ACM*, 46, 6, 31.
5. Goldfarb, C. F., and Prescod, P. (1999) The XML handbook*,* Prentice Hall PTR Upper Saddle River, NJ, USA.
6. Gottschalk, K., Graham, S., Kreger, H., and Snell, J. (2002) Introduction to Web services architecture, *IBM Systems Journal,* 41, 2, 170-177.
7. Hill, N. C., and Ferguson, D. M. (1989) Electronic Data Interchange: A Definition and Perspective, *EDI Forum: The Journal of Electronic Data Interchange*, 1, 1, 5-12.
8. Hix, D., Gabbard, J., Swan, E., Livingston, M., Hollerer, T., Julier, S., Baillot, Y., and Brown, D. (2004) A Cost-Effective Usability Evaluation Progression for Novel Interactive Systems, *Proceedings of the 37th Hawaii International Conference on System Sciences* (HICSS).
9. Hayes, J., Peyrovian, E., Sarin, S., Schmidt, M., Swenson, K., and Weber, R. (2000) Workflow Interoperability Standards for the Internet, *IEEE Internet Computing*, 4, 3, 37-45.
10. Khoumbati, K., Themistocleous M., and Irani, Z. (2005) Integration Technology Adoption in Healthcare Organizations: A Case for Enterprise Application Integration, *Proceedings of the 38th Hawaii International Conference on System Sciences* (HICSS), January, 3-6, Hawaii, USA, 149a- 149a.

11. Kim, Y., Kang, S., Kim, D., Bae, J., and Ju, K. (2000) WW-FLOW: Web-Based Workflow Management with Runtime Encapsulation, *IEEE Internet Computing*, 4, 3, 55-64.
12. Kreger, H. (2001) Web Services Conceptual Architecture (WSCA 1.0). *IBM Software Group*, 5, 6-7.
13. Kwak, M., Shim, J., and Han, D. (2002) A Framework Supporting Dynamic Workflow Interoperation and Enterprise Application Integration, *Proceedings of the 35th Hawaii International Conference on System Sciences* (HICSS), January, 7-10, Hawaii, USA, 3766- 3775.
14. Muehlen, M. (2004) Workflow-based Process Controlling: Foundation, Design, and Application of Workflow-driven Process Information Systems, Logos Verlag, Berlin.
15. Object Management Group (2000) Workflow Management Facility Specification, Version 1.2.
16. PESC (Postsecondary Electronic Standards Council), Data Transport Standard V 1.0 Reference Implementation Guide, http://www.pesc.org/info/docs/data_transport/Data%20Transport%20Standard%20v%201_0%20Reference%20Implementation%20Guide.doc.
17. PESC (Postsecondary Electronic Standards Council), EDI Implementation Guide, http://www.pesc.org/info/implementation-guides.asp.
18. Schmidt, M. T. (1999) The Evolution of Workflow Standards, *IEEE Concurrency*, 7, 3, 44-52.
19. Sollazzo, T., Handschuh, S., Staab, S., and Frank, M. (2000) Semantic Web Service Architecture - Evolving Web Service Standards toward the Semantic Web, *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference,* May 22-24, 2000, Orlando, Florida, USA, 425-429.
20. W3C (2004) Web Services Architecture, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/