Research on E-business Intelligent Examination System

Xin Jin
School of Information
Central University of Finance& Economics
Beijing, 100081, P.R.China

Abstract: - Commonly, the traditional E-business education and training adopts two ways: the theoretics teaching in classroom and the computer-based application operation in E-business simulation environment which is like an E-business web site (but not same as). But it is difficult to check students' ability of E-business application operation. In this paper, we will discuss an intelligent examination system that is for checking the examinee's ability of E-business application operation. This paper presents the design and implementation of an intelligent examination system for checking the E-business application operation capable of the examinees. It proposes a novel architecture for on-line examination system which is based on E-business workflow. That system adopts the common client-server pattern with two major parts, Student site and Teacher site. The Student site provides examinee a GUI (Graphics User Interface) embedded a WWW browser for answering the paper. The Teacher site mainly includes paper management subsystem for managing paper database, a building paper subsystem for building a new paper, and a scoring paper subsystem for scoring the paper of the examinees. In this paper, we also analyze and discuss the system architecture, the system simulation environment, some key questions and the corresponding solutions.

Key-Words: - intelligence, intelligent examination system, E-business application operation, workflow, E-business simulation environment

1 Introduction

As we all known, the Information Technology (IT) is the important support for E-business. The use of information technology has become a primary survival factor for business organizations in a global competitive environment, as the E-business tide is spreading through various domains violently in modern society. Developing E-business is an important factor to accelerate national economy increasing.

In order to adapt and grasp this situation and respond the proposal in the Eleventh Five Planning of Nation "accelerate developing E-business, apply information technology widely and popular information knowledge and craftsmanship widely in the whole society.". So the education and the training of the E-business knowledge are actively developed in whole society. Under that great background, we develop the Examination System for E-business Application Operation (ESBAO) which is a part of E-business education and training software system supported by Shanghai Informatization Office, so as to check the user's ability of the E-business application operation.

Commonly, the traditional E-business education and training adopts two ways: the theoretics teaching in classroom and the computer-based application operation in E-business simulation environment which is like an E-business web site(but not same as). Here, We will introduce examination system that is for checking the examinee's ability of E-business application operation. This system has some characters as follows:

I. Checking the ability to E-business application operation of the examinees based on E-business workflow

The ESBAO system gives marks for the students according to whether their operations which include the key operation steps and the operational results are correct or not under the E-business simulation environment. For example, if the question is buying commodity A by the searching way, you can do the correct answer as following workflow: *start up searching engine -> select commodity A -> start up purchasing engine -> make orders -> finish*. Your answer also can be completed through the other workflow. This system gives marks for you just according to whether starting up searching engineer and successfully purchasing A or not.

II. Intelligence

The intelligence of ESBAO is embodied mostly with the intelligence of building examination paper subsystem and scoring examination paper subsystem in Teacher site. For example, the teacher can building

the examination paper by himself through GUI, also he can only start up the Building Paper Agent (BPA) which can build examination paper automatically. When the teacher scores the examination papers, he only needs to star up scoring paper engine through GUI, then the system would check and mark the examination papers and record the examinees' scores in the database.

The rest of the paper is organized as follows: In section 2 we introduce the design of the ESBAO system architecture. In section 3 discuss the system design and implementation essential. In section 4, we discuss the key technologies for system implementation. Finally it is the conclusion of this paper.

2 System Architecture

The ESBAO is based on E-business simulation environment, and it use the XML technology, the Component technology, the Database Trigger and Stored Procedure technology and other information technology to design and realize the ESBAO. That system adopts the common client-server pattern with two major parts, the client component also called Student site and the application-oriented server component here also called Teacher site. The Student site provides examinee a GUI (Graphics User Interface) embedded a WWW browser for answering the paper. The Teacher site mainly includes paper management subsystem for managing paper database, a building paper subsystem for building a new paper, and a scoring paper subsystem for scoring the paper of the examinees. The system also includes the Paper Database and some intelligent toolkit such as Paper Creator, Building Paper Agent, Monitor for monitoring application operations of the examinees, Answer Generator and Scoring Paper Generator. The Fig.1 depicts the system architecture and the workflow.

In Fig.1, the Paper Database is collector of examination questions described by XML document style. The teacher can simply and quickly creates a new XML document paper, only need to start up Paper Creator component engine that can work automatically. The Paper Creator is very expediently for the teacher to parser XML document paper to automatically analyze and deal the paper's structure, attributes, scores, contents etc. When the teacher finishes building paper, the corresponding standard answers are automatically created. These building paper procedures also can work automatically by Building Paper Agent, after the teacher start up the agent.

When the examinees answer the questions of the paper, it is necessary to operating in the E-business simulation environment which is like but not same as the E-business web site. The main difference is that E-business simulation environment sets a monitor program in some main operation steps to capture who and whenever and however to operating. The monitor program is oriented special operation workflow, and it is sequential and discretely in workflow and in work time, so as to the system records the operations which are the some key operation steps or result in database. When scoring the papers, we can use Answer Generator to transfer the recorders of the database to XML formatted answer sheets. In this system, all of documents are XML-based format.

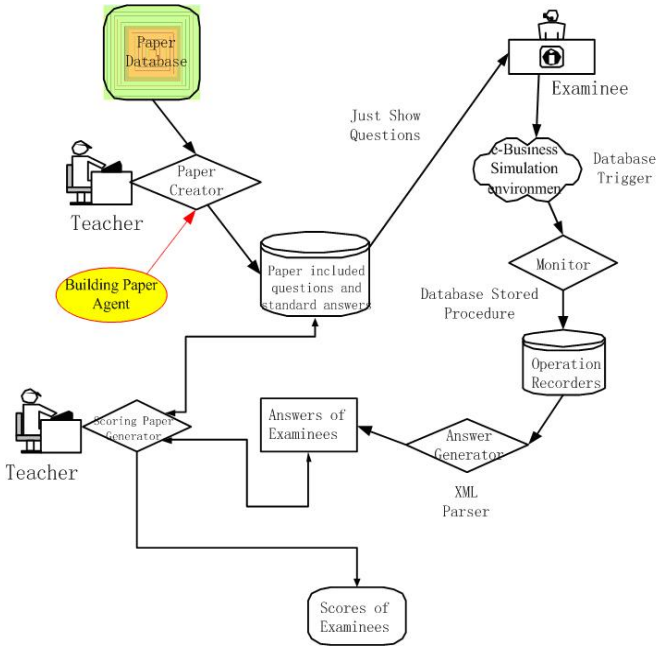

Fig.1 System Architecture and Workflow of the ESBAO

When the teacher starts up the Scoring Paper Generator which can score papers automatically, it can compare examinees' answers with the standard answer to give marks. Because the all document are uniform XML format, it only needs to judge the data identity, then to mark the scores.

3 E-business System Simulation Environment

The E-business simulation environment is an important part of the ESBAO system to support the whole system to run.

3.1 Introduction to E-business Environment

E-business applications operations are always in a dynamic and distributed environment, dealing with a large number of heterogeneous information sources with evolving contents and dynamic availability. They typically rely on distributed and autonomous tasks for information search, fusion, extraction and processing, without centralized control. E-business typically involves the following activities: identifying requirements; brokering products; brokering vendors; negotiating deals; or making purchase and payment transactions. Today, these activities are initiated and executed by humans. In the future, we see them being conducted by software agents. Software agents are personalized, continuously running and semi-autonomous, driven by a set of beliefs, desires and intentions (BDI). They can be used to mediate users and servers to automate a number of the most time-consuming tasks in E-business, with enhanced parallelism. Agents can also be used for business intelligence, such as discovering patterns (e.g. shopping behavior patterns or service providing patterns) and react to pattern changes. For example, suppose the sales of VCRs had been strongly associated with the sales of TVs, but this association has recently weakened as TV buyers turn to buying DVDs instead of VCRs. Such a change in the association helps to explain or predict the slow down of VCR sales. Moreover, agents can selectively preserve data and themselves become dynamic information sources.

E-business is also a plug and play environment. Business processes and agent cooperation are embedded in each other. Services need to be created dynamically on demand. Business partnerships (e.g. between suppliers, resellers, brokers, and customers) need to be created dynamically and maintained only for the required duration such as a single transaction.

The dynamic nature of E-business requires multi-agent cooperation to be based on dynamic ontology. By dynamic ontology we mean that the concepts, rules and facts underlying agent interaction, are different from domain to domain, and vary from time to time. In order to automate agent cooperation, it is necessary to provide a standard format for encoding messages with meaningful structure and semantics, as well as domain ontology that agents can readily exchange and interpret. This format should be common for agent communication as well as for E-business data exchange in general. The extensible markup language, XML, is becoming the standard for data interchange on the Web. We use XML for the above purpose. Business processes, or workflows, may be considered as a kind of multi-agent cooperation, in the sense that software agents may be used to perform tasks of business processes, and workflow may be used to orchestrate or control the interactions between agents. We envisage the need for dynamically plugging them into each other.

3.2 Dynamic Agent

We have developed a c# based dynamic agent infrastructure for E-business which supports dynamic behavior modification of agents, a significant difference from other agent platforms. A dynamic agent does not have a fixed set of predefined functions, but instead, it carries application- specific actions, which can be loaded and modified on the fly.

A dynamic-agent has a fixed part and a changeable part. As its fixed part, a dynamic-agent is provided with light-weight, built- in management facilities for distributed communication, object storage and resource management. A dynamic agent is capable of carrying data, knowledge and programs as objects, and executing the programs. The data and programs carried by a dynamic agent form its changeable part. All newly created agents are the same; their application-specific behaviors are gained and modified by dynamically loading Java classes representing data, knowledge and application programs

These capabilities allow a dynamic agent to adjust its capabilities and play different roles to accommodate changes in the environment and requirements. Through messaging, dynamic agents can expose their knowledge, abilities and intentions, present requests and exchange objects. They can move to the appropriate location for high-bandwidth conversation. They can also manage their own resources across actions. Such an infrastructure supports dynamic service construction, modification and movement, and allows a dynamic agent to participate in multiple applications and dynamically formed partnerships. With these features, dynamic agents fit well into the dynamic E-business environment. A multi-agent cooperation infrastructure is developed for E-business automation, where dynamic agents perform various market activities, cooperating through exchanging data as well as programs, switching roles and forming dynamic partnership that exists only when needed. For example, the agents reselling products, the agents supplying products and the agents providing brokering services for negotiating service terms, etc, may form dynamic partnership for a specific business application. In this way, dynamic agents cooperatively support plug-and-play commerce, allowing businesses to be built on one another's services.

3.2 XML Messaging

Autonomous agents cooperate by sending messages and using concepts from a domain ontology. A standard message format with meaningful structure and semantics, and a mechanism for agents to exchange ontologies and message interpreters, have become key issues. Furthermore, the message format should be accepted not only by the agent research community, but also by all information providers. Dynamic agents send and receive information through XML encoded messages. We use a KQML/FIPA ACL-like format, encoded in XML. In fact, an XML document is an information container for reusable and customizable components, which can be used by any receiving agent This is the foundation for document-driven agent cooperation. By making Web accessible to agents with XML, the need for customer interfaces for each consumer and supplier will be eliminated. Agents may use XML format to explain their BDI, explaining new performatives by existing, mutually understood ones. Based on the commonly agreed tags, agents may use different style DTDs to fit the taste of the business units they mediate. Further, a dynamic agent can carry an XML front-end to a database for data exchange, where both queries and answers are XML encoded. The power of XML, the role of XML in E-business, and even the use of XML for agent communication, have been recognized. Although XML is well structured for encoding semantically meaningful information, it must be based on an ontology. As ontology varies from domain to domain, and dynamic for dynamically formed domains, The more significant issue is to exchange the semantics of domain models, and interpret messages differently in different problem domains. Generally speaking, domain ontology provides a set of concepts, or meta-data, that can be queried, advertised and used to control the behavior of agent cooperation. These concepts can be marked using XML tags, and then a set of commonly agreed tags, underlie message interpretation. The structures and the semantics of the documents used in a particular problem domain are represented by the corresponding DTDs and interpreters. We use different languages, all in XML format, for different problem domains, such as product ordering, market analysis, etc. Accordingly, we use an individual interpreter for each language. Dynamic agents can exchange those DTDs together with documents, and exchange those interpreters as programming objects, in order to understand each other in communication. These approaches allow us to provide a unified application carrier architecture, a unified agent communication mechanism, a unified way of data flow, control flow and even program flow, but flexible application switching capability, for supporting E- business.

3.2 Workflow Engine

The workflow engine in E-business environment provides flow control for business process automation. Business processes often involve multilevel collaborative and transactional tasks. Each task represents a logical piece of work that contributes to a process. A task at leaf-level is performed by a role. A role is filled at run-time with a user or a program. A process and its tasks are handled at separate layers. At the process layer, centralized coordination is supported; and at the task layer, location distribution, platform heterogeneity and control autonomy, are allowed. Business processes may be considered as a kind of multi-agent cooperation, in the sense that a software agent can be used to fill a role for performing a task in a workflow, and workflow can be used to orchestrate or control the interactions between agents. However, many related activities in E-business automation do not form synchronized, traditional workflow, but requires more dynamic agent cooperation. In order to combine the strength of workflow and agent cooperation for supporting E-business, it is necessary to understand their relationship and difference. We have developed the mechanisms for plugging workflow in agent cooperation, and plugging agent cooperation in tasks of business processes, are introduced. In particular, dynamic workflow service provisioning is supported, allowing workflow servers to be built on the fly. E-business is a dynamic, distributed and a plug and play environment for which we expect software agent based technologies to become increasingly important. However, since agents with static capability cannot dynamically load new functions, cannot change their predefined behavior, and cannot exchange programs with others, they are unable to switch roles, to participate in multiple applications, or to be involved in dynamically formed partnerships. Therefore, static agent frameworks are not really suitable for the highly dynamic E-business applications.

4 Discuss the System Design and Implementation Essentials

The ESBAO system can check the examinee's application operation procedures and results under the E-business simulation environment. Its designing goal is to test E-business application operation ability. It is necessary to solve some important questions as follows:

- *capturing and analyzing the operations*

- *concurrency of many operations*
- *building paper and scoring paper automatically*
- *capturing and analyzing dynamic workflow*

4.1 Capturing and Analyzing the Operations

The most methods to recording the operation procedures are relative with the structure of the E-business simulation environment. If it is Client/Server structure, the ESBAO system can set monitor program in client and server together. If it is Browser/Server structure (this system uses this structure in fact), the system can only set monitor program focused in server to search and capture the operations of the client. As the above mentioned, this monitor methods are discretely in the business workflow. So before building monitor program centralized in server, we must firstly be confirmed the monitored objects. That means we need to confirm which operation steps and results need to be monitored.

The simply solution is to add triggers in database to monitor add, delete and update operations. These data changing often be from the activity and the operation in business environment. Furthermore, because the system design is based on server component program, we can add log records in component to capture operations. We can add an examination monitor switcher for all components. When beginning the examination, the system open this switcher, then the operations and activities can request these components to create some fixed format records which can reflect the operation types.

Furthermore, if some contexts of simulation environment has used middleware or middleware platform, we can set a series of monitor program to realize the operation capturing and analyzing.

4.2 Concurrency of Many Operations

At current, the concurrency is not very difficult to implement contrasted to past time, because the most of commerce software and the system program developing architecture provide responding solution. But in the ESBAO system, because there are many examinees to take part in examination together in a server E-business simulation environment, so there are some questions to solve as follows. For example:

- When we operate a certain defined resource, if the resource is not database (the database can balance the concurrent operation) but data file, media or print device, Once the resource be locked by one student's operations, the other students can not use that resource to finish the operation. Besides the later operation could overlap the previous operation, so some different meanings are arose.

- In some typical E-business application workflow, some enterprise entity application, especially the application come down to financing, auction, sale or paying etc., often be sensitive to the activity entity. For example, auditing the order, the auditing order can not be looked through by other people. The kind of operations, such as the above operation of auditing order, can not fit as the examination questions.

- Besides some E-business application workflows often need at least two aspects to finish. For example, the online auction in which there are some business entity which often affect and restrict each other. So, to check many people cooperation operation is an interesting approach issue.

In this system, we can adopt the following ways to solve the above concurrent questions.

- *Adopt memory database and data view.*

In the common situation, the data of the examinee operation often distribute in temp data cache and view but not operate directly in the center database. Then the data of examinee operation can not be interfered each other.

- *Isolate the public data and private data.*

In some typical workflow, the system can create some backups of used data and resource. Different examinees can use different backups. It is not need to backup the data which is not relative with these operations.

- *Using programs to simulate many people cooperation operation.*

For example, in the contesting auction, the system adds the dummy contest rival to contest with the examinee.

- *Data restore.*

After the examination, in order to keep the stability and standardization, the system provided toolkits which can restore the data and the resource to the primal status.

4.3 Manage Papers and Score Papers Automatically

In order to manage the papers, using the paper database is a good solution. In the database, all examination questions have been test to be sure the correctness in the examination procedure. To the examination in simulation environment based on the components and comprised of some work units, the teacher can use the papermaking toolkit to build the questions and do parameter setting in operation units of the integrate workflow to form a series of operation rules. The component and the work units can judge the mark referring those rules, so those methods can make the questions type and formal to be more diversified.

In the ESBAO system, besides the checking paper automatically based on XML documents compared, we add the personality requirement and evaluating guidelines, which make the marking automatically to be more impersonality. In most situations, E-business operations can not record and weight by estimated way. For example, using the searching engine, we can search one commodity through various ways. In that procedure, it is difficult to judge the operation is correct or not. Those lead to the process-typed questions and result-typed questions appeared. The called process-typed question means to emphasize the sequence of a series of operations, but the result-typed questions only emphasize the final result.

For those two different questions types, we need use different judging standards to deal. So in the process-typed questions, we can set some monitor points in the operation procedure, also we can divide the integrated workflow to many sub objects. The system not stickle to the real application workflow, but add more require and evaluation guidelines artificially. This mark based on the subdivision workflows is apparently more correct.

4.4 Capturing and Analyzing Dynamic Workflow

Business processes that operate within, across or between organisations in order to implement value chains that can be used to deliver E-business transactions may be implemented using a set of workflow definitions that have been created to support discrete segments of the overall process. This scenario poses the question of how to avoid creating islands of automation in the operation of an end to end business process. The answer to this problem is workflow interoperability – the enabling of different workflow products to "talk to each other" by exchanging messages that effect process interoperation and integration to drive and manage the operation of the value chain. Workflow interoperability enables the owner of the value chain to have greater visibility and control over its performance and participants within the value chain benefit from flexibility, and improved control and visibility over the performance of the processes they operate and the processes with which they interoperate.

In E-business environment, there are some dynamic business activities such as the auction in C2C E-business model. It is difficult to capture the dynamic business workflow. We need to link the different E-business entities to form the E-business workflow chain according to the E-business operation activities. The complex E-business activity always be composed by some simply activities. The basic activity is the *request and response* in the dynamic workflow, such as the *purchasing* activity which is composed by many request and response activities between the buyer and the seller in B2C E-business model.

We use the Web Services technology to describe and capture the E-business dynamic activity. Web Services utilize some XML-based standards: Simple Object Access Protocol (SOAP), Universal Description, Discovery, and Integration (UDDI), and Web Services Description Language (WSDL), and several other protocols that allow the applications to communicate with others. XML has become the standard for defining data interchange formats on the internet. SOAP is an XML-based protocol for exchanging information in a decentralized, distributed environment. It provides an envelope that defines a framework for describing what is in a message and how to process it, encoding rules for expressing application-defined datatypes, and a convention for representing remote procedure calls and responses. UDDI is an XML-based specification for a registry of businesses and the Web Services they offer. By providing the necessary translations, it enables software to automatically discover Web Services and integrate with them. WSDL lets developers expose the syntax of a Web Service. Using XML format, it describes network services as a set of endpoints operating on messages containing either document or procedure-oriented information, and the operations and messages are described abstractly and then bound to a concrete network protocol and message format to define the endpoints.

A simple Web Services is characterized by the three XML-based standards SOAP(Simple Object Access Protocol), UDDI(Universal Description, Discovery and Integration) and WSDL(Web Services Description Language), which taken together to provide a basic "request and response" functionality. The complex Web Services might involve multipart, dynamic-running transactions. It

is easy to link different E-business entities to form E-business workflow by using Web Services. Therefore we use Web Services to capture and analyze dynamic business workflow in ESBAO system.

5 Paper Database based on Ontology

Ontology is defined as a formal and explicit representation of a conceptualization. It allows one to represent, in a more or less formal way, concepts of a domain of concern and their relations. An ontology is described using attributes, properties, relations between concepts, and eventually constraints and axioms. An ontology can be used to provide a formal and shared understanding of some domain, facilitating exploitation by both human agents or computer programs. In this section, we will discuss how to build and manage the papers based on the ontology.

5.1 Paper Ontology with XML/XML Schemas

XML (Extensible Markup Language), the meta language developed by the World Wide Web Consortium (W3C) for information structuring and exchange on the Web, seems a priori to be a good candidate to describe ontologies in a distributed environment. XML allows the definition of customized markup languages with application-specific tags, e.g., <QUANTITY> or <SPEED>, for representing information in particular application domains and defining data structures. XML Document Type Definitions (DTDs) provide a way to explicitly declare the tag sets and their structure, to be used in particular units of data. The advantages of XML are that it has a human-readable and well-defined syntax. Furthermore, there exist software tools for parsing and manipulating XML. However, even if XML allows the specification of user-defined tags, it does not provide the semantics required for an ontology. What does the <RESOURCE> tag mean? In which unit is a <SPEED> concept represented? Furthermore, even if DTDs define the legal nestings of tags in a document, it does not represent the notion of an ontological class hierarchy. So, the inheritance mechanism is missing. The description of ontologies requires ways to explicitly specify relations between concepts, hierarchies of concepts, in order to offer more expressiveness. So, new proposals are emerging to address this aspect on top of the XML language, for example XML and RDF schemas. These meta-models should be considered for the construction of coalition ontologies.

Due to several DTDs limitations, in particular DTDs are not adequate for describing data contents, XML Schemas is a new W3C proposal aimed to replace DTDs. XML schemas are built in XML, provide data types as well as relationships between elements, and support namespaces. However, XML schemas present limitations for object-based knowledge representation, in particular the lack of inheritance that is necessary for ontology representation. The Fig.2 shows the paper ontology with XML SHCEMA

```
<PAPER>
  <PAPER_NO></PAPER_NO>          //Paper ID
  <PAPER_DATE></PAPER_DATE>      //Testing Date
   <TEST EX_ID="" >
     <T_TITLE> </T_TITLE>   //The Question
    <T_SCORE> </T_SCORE>   //Score Value
     <T_QUESTION  T_ID="1">//The Selective Item
     </T_QUESTION>
     <T_QUESTION  T_ID="2"></T_QUESTION>
                . . . . . .
     </T_QUESTION>
      <T_KEY> </T_KEY>//The Correct Answer
    </TEST>
    .......
</PAPER>
```

Fig.2 The paper Ontology with xml/xml schema

5.2 Ontology and Database Schemas

Ontologies and database schemas are closely related and people often have trouble deciding which is which. There is often no tangible difference, no way of identifying which representation is a schema and which is an ontology. This is especially true for schemas represented using a semantic data model. The main difference is one of purpose. An ontology is developed in order to define the meaning of the terms used in some domain whereas a schema is developed in order to model some data.

Although there is often some correspondence between a data model and the meaning of the terms used, this is not necessarily the case. Both schemas and ontologies play key roles in heterogeneous information integration because both semantics and data structures are important. For example, the terminology used in schemas is often not the best way to describe the content of a resource to people or machines. If we use the terms defined in a resource ontology to describe the contents of a resource, queries that are sent to the resource will also use these terms. In order to answer such queries, there needs to be a relationship defined between the ontology and

the resource schema. Again, declarative mappings that can be interpreted by some mediator system are useful here. The structural information provided by schemas will enable the construction of executable queries such as SQL queries. This is related to the discussion earlier about XML, where a database schema is analogous to an XML schema or DTD. As pointed out above, using XML is insufficient for determining the semantics of resources. A schema, whether specified using XML or some database schema language, needs an associated formal ontology in order to make the semantics of the resource clear. When the meaning of data and schemas is made explicit using an ontology, programs can be designed that exploit those semantics.

5.3 Entity Correspondence

Ontologies are used in E-business environments where data is scattered across heterogeneous distributed systems. In order for the consumer to have access to the maximum amount of available information, we want to be able to retrieve information from various systems and to integrate it. For example, we might want to integrate information from a supplier's product catalogue with customer reviews produced independently.

To gather all the information relevant to an entities, the correspondence between entities across resources must be established. For example, the academic records and criminal records of a person are likely to be stored in separated data resources. However, the way in which different resources identify individuals varies. For example, in relational databases entities are identified using key attributes. There is no guarantee that different relational databases use the same key attributes. Even when the same key attribute is used, different terms may be used to denote the attributes.

6 Key Technologies for System Implementation

On the above discussion, we analyzed the system design requirement. Because of the characters of testing contexts and the intelligence requirement, we adopt some technologies to implement the system as follows:

- *Component technology*

Considered the characters of the operation mode, the design based on components gives priority of business- oriented or procedure- oriented. In order to combine conveniently the script language of simulation environment (in this system, our simulation environment adopted the ASP script), we use the C++ and VB to compile the COM we needs. For example, in order to record the examinees' answers, we need to set monitor program to monitor that operation procedure. In ESBAO system , we use the component technology to enclose most functions to strengthen the software reused.

- *Ontology and XML technology*

In this system, we use ontologies based on XML Schema to represent the paper and other E-business knowledge. Moreover we use XML-based documents to store the data of the papers, and use XML technology, such as XML DOM ,XML SAX and ADO, to realize the dynamically mapping from the XML documents to database recorders, transferring information and maintaining information etc.

- *Web Services*

Web Services is characterized by the three XML-based standards SOAP, and WSDL, which taken together to provide a basic "request and response" functionality. Thus the Web Services can be used to efficiently deliver information between one business entity and the other in E-business environment. The Web Services can link E-business activities to form E-business workflow. We use Web Services technology to describe, capture and analyze dynamic business workflow.

- *Database technology*

One hand the database is the carrier of the data storage, the other hand the system can use the database triggers and/or the component events to stimulate the database stored procedures to record the operations of the examinees.

- *Agent technology*

In this system, the Building Paper Agent is an automatic component in fact, in which we add the message mechanism to realize the autonomy of the agent. We use C++ to implement the agent.

7 Conclusion

In this paper, we discuss an operation-oriented intelligence examination system, which is for E-business application operation examination, and present a novel system architecture for the Examination System for E-business Application Operation (ESBAO). The ESBAO system mainly include an answering paper subsystem for examinees in the Student site, paper management subsystem in Teacher site for managing paper database, a building paper subsystem in Teacher site for building a new paper, and a scoring paper subsystem in Teacher site for scoring the paper of the examinees. In this paper, we detailedly discuss the system structure, the

E-business simulation environment, some key questions and the corresponding solution.

Acknowledgement

This research was supported by Natural Science Foundation of Beijing under Grant No. 4073041.

References:

[1] Blex,udC, *XML Technology,* Beijing: Tsinghua University Press, 2003

[2] Mary Kirtland, *Application Program Based on COM* ,Beijing: Peking University Press, 1999

[3] Lucian Popa，Yannis Velegrakis, *Translating Web Data,* Proceeding of the 28th VLDB Conference,HongKong China, 2002

[4] Maurizio Lenzerini, *Data integration: A theoretical perspective,* In PODS, 2002, pp. 233-246

[5] Jeffrey .C, *Web Services Architecture Usage Scenarios W3C Working Draft[*EB/OL], http://www.w3.org/TR/2002/WD-ws-arch-scenarios-20020730/ , July.30 2002

[6] Zhongzhi Shi, *Intelligent Agent and Application:* Beijing: Science Technology Press, 2001

[7]Olivier Perrin , Claude Godart, *A model to support collaborative work in virtual enterprises*, Data & Knowledge Engineering, v.50 n.1, p.63-86, July 2004

[8]M. Brian Blake , Hassan Gomaa, *Agent-oriented mpositional approaches to services-based cross-organizational workflow*, Decision Support Systems, v.40 n.1, p.31-50, July 2005

[9]Jorge Cardoso , Robert P. Bostrom , Amit Sheth, *Workflow Management Systems and ERP Systems: Differences, Commonalities, and Applications*, Information Technology and Management, v.5 n.3-4, p.319-338, July-October 2004

[10]Q. Chen, P. Chundi, Umesh Dayal, M. Hsu, "*Dynamic-Agents*", International Journal on Cooperative Information Systems, 1999, USA

[11]Botelho, L. and Ramos, P. (2000) "*Extending the FIPA ACL language: From object based descriptions to relational representations*". In Proceedings of the 3rd Iberoamerican Workshop on Distributed AI and Multi-Agent Systems (DAIMAS 2000)

[12]Berners Lee, T., *Semantic Web Road map*, September98, http://www.w3.org/DesignIssues/Semantic.html

[13]V. Chandhri, A. Farquhar, R. Fikes, P. Karp and J. Rice. *OKBC: A Programmatic Foundation for Knowledge Base Interoperability*. In Proceedings of AAAI-98, July 26-30, Madison, WI.

[14]Stephen Cranefield. *UML and the Semantic Web*. In Proceedings of the International Semantic Web Working Symposium (SWWS2001), 2001. http://www.otago.ac.nz/informationscience/pubctns/complete/papers/dp2001-04.pdf.gz

[15]Cranefield, S., Nowostawski, M., and Purvis, M. (2001) "*Implementing agent communication languages directly from UML specifications*". Discussion Paper 2001/03, Department of Information Science, University of Otago, New Zealand http://www.otago.ac.nz/informationscience/pubctns/complete/papers/dp2001-03.pdf.gz

[16]Cranefield, S. and Purvis, M. (2000) "*Extending agent messaging to enable OO information exchange"*. In Cybernetics and Systems 2000, Proceedings of the 5th European Meeting on Cybernetics and Systems Research (EMCSR 2000), Trappl, R., editor (Vienna. Austrian Society for Cybernetic Studies)

[17]Guido L. Geerts and William E. McCarthy. *An accounting object infrastructure for knowledge-based enterprise models*. IEEE Intelligent Systems, July/August, 1999

[18]N.Guario. *Formal Ontology, Conceptual Analysis and Knowledge Representation*. Special issue of the International Journal of Human and Computer Studies, vol. 43 n. 5/6, Academic Press,1995.

[19]N. Guarino. *Formal Ontology and Information Systems*. Proceedings of FOIS'98. Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp.3-15

[20]Haustein, S. (2000) "*rdf for object serialization"*. *Start of thread on www-rdf-interest mailing list.* http://lists.w3.org/Archives/Public/www-rdf-interest/2000Feb/0157.html

[21]J. Heflin and J. Hendler. *Dynamic Ontologies on the Web*. American Association for Artificial Intelligence, 2000

[22]R. Hull, A. Kumar and J. Simeon. *Smart Supply Web: An Application of Web-based Data and Workflow Mediation*. http://www-db.research.bell-labs.com/user/simeon/workshop.pdf

[23]Jagannathan, V. and Fuchs, M. (1999) *Workshop report on integrating XML and distributed object technologies"*. In 8th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE'99) (IEEE Computer Society Press) http://www.jeffsutherland.org/xml/IEEE_XML_Report_Draft.htm

[24]Y. Iwasaki, A. Farquhar, R.Fikes and J. Rice. *A Web-Based Compositional Modeling System for Sharing of Physical Knowledge.* In Proceedings of The Fifteenth International Joint Conference on Artificial Intelligence. 1997. Nagoya, Japan, pp.494-500.

[25]J.Jannink, S. Pichai, D. Verheijen and G. Wiederhold. *Encapsulation and Composition of Ontologies.* In Proceeding of AAAI Workshop on AI & Information Integration,1998. http://citeseer.nj.nec.com/jannink98encapsulation.html

[26]Klein, M., Fensel, D., van Harmelen, F., and Horrocks, I. (2000) *"The relation between ontologies and schema-languages: translating OIL-specifications in XML-Schema"*. In Proceedings of the Workshop on Applications of Ontologies and Problem-Solving Methods, 14th European Conference on Artificial Intelligence (ECAI 2000) http://delicias.dia.fi.upm.es/WORKSHOP/ECAI00/7.pdf

[27]N. Krishnakumar and A. Sheth. *Managing Heterogeneous Multi-system Tasks to Support Enterprise-wide Operations.* Distributed and Parallel Databases, 3, 1-33(1995). Kluwer Academic Publishers, Boston, 1995.

[28]K. Liu and V. Tsaoussidis. *Efficient Network Management for Collaborative Services and Application Development.* International Conference on Multimedia and Expo (I)", pp.53-56, 2000.

[29]D. Maluf and G. Wiederhold. *Abstraction of Representation for Interoperation.* In Proceeding of International Syposium on Methodologies for Intelligent Systems 1997, pp.441-455, 1997.

[30]S. Mellraith, T. Son and H. Zeng. *Semantic Web Services.* IEEE Intelligent Systems, March~April, 2001, pp.46~53.

[31]E. Mena, A.Illarramendi, V. Kashyap and A. Sheth. *OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies.* Kluwer Academic Publishers, Boston. 1998.

[32]Sergey Melnik, Stefan Decker. *A Layered Approach to Information Modeling and Interoperability on the Web*, http://www-db.stanford.edu/~melnik/pub/sw00/sw00.pdf

[33]OMG (2000a) *"Meta Object Facility specification version 1.3"*. Object Management Group http://www.omg.org/technology/documents/formal/meta.htm

[34]OMG (2000b) *"XML metadata interchange specification, version 1.0"*. Object Management Group http://www.omg.org/technology/documents/formal/xml_metadata_interchange.htm