E-learning-Oriented Software Architecture Design and Case Study

http://dx.doi.org/10.3991/ijet.v10i4.4698

Peng Lu[1], Xiao Cong[1], Dongdai Zhou[2]
[1] Northeast Dianli University, Jilin, Jilin, China
[2] Northeast Normal University, Changchun, Jilin, China

***Abstract*—Nowadays, E-learning systems have been widely applied to practical teaching. They are favored by people for their characterized course arrangement and flexible learning schedule. However, the system does have some problems regarding application, e.g. the functions of a single software are not diversified enough to completely satisfy the requirements for teaching. To cater to more applications in the teaching process, it is necessary to integrate functions from different systems. But the difference in developing techniques and the inflexibility in design make implementation difficult. The major reason for these problems is the lack of fine software architecture. In this article, we present a domain model and component model of an E-learning system and component integration method on the basis of WebService. And we proposed an abstract framework of E-learning that could express the semantic relationship among components and realize high level reusability on the basis of an informational teaching mode. Based on this foundation, we formed an E-learning oriented layering software architecture containing a component library layer, application framework layer and application layer. Moreover, the system contains layer division multiplexing and was not built on developing language and tools. With the help of the software architecture, we built a characterized flexible E-learning system through framework selection, component assembling and replacement. In addition, we show how to build a concrete E-learning system on the basis of this software architecture.**

***Index Terms*—Software Architecture, Component, Web Service, Framework.**

I. INTRODUCTION

E-learning is a teaching method based on digital techniques [1]. Contrary to a traditional teaching mode, the E-learning based mode provides convenient and a flexible interactive learning environment for teachers and students as well as among students. Its virtue lies in the activity for learning behavior, transcendence of time and space, interactivity, openness, etc. These are important for development of students' advanced cognitive ability and improvement of their active learning ability as well as for teaching quality.

To implement E-learning, a digital learning environment is needed, which means the E-learning system must support the teaching and learning activities. Behaviorism, cognitivism and constructivism provide theoretical foundations for E-learning system design from different perspectives [2].

a. A behaviorist-based strategy could be used for factual teaching (what).

b. A cognitivist-based strategy could be used in principle and process teaching (how).

c. A constructivist-based strategy could be used in the learning of knowledge about actual life and circumstances, etc.

So far, there are many E-learning systems like Blackboard, Moodle, sakai and Atutor. Although these E-learning systems have been widely applied in elementary and secondary schools and colleges, two problems exist:

a. A single system is not comprehensive in function, furthermore, the system structure is fixed and the function cannot be customized. So, it is difficult for the system to support all of the teaching and learning modes; it is especially difficult to completely satisfy all the needs for teachers' individualized teaching and learners' individualized learning activities. For example, a Blackboard platform cannot support all the research teaching modes, especially the WebQuest-based teaching mode.

b. To satisfy the application in teaching and learning, it is necessary to integrate systems or parts of the function modules of the system that can support different teaching modes. However, different developers used different software implementation techniques to design and develop the E-learning system, and different parts of the system are coupled. So, it is difficult to integrate existing function modules or application programs into other application programs or to integrate the function module of other systems or application programs into existing systems.

One of the main reasons for the above problems is that the design and development of the software system is not reasonable or perfect enough. Although behaviorist, cognitivist and constructivist theories provide a basis for the design and development of E-learning, from the perspective of the software function requirement, there are no instructions in the level of the system implementation techniques. From the perspective of software engineering, a software system that works well is not only determined by what functions it can provide but also by its non-functional factors, such as adaptability (customization of the system), reusability (tendency of weaving and reconstructing the system), transferability (tendency of weaving of system) and reliability.

These non-functional factors are determined by the software architecture of the system [3]. So to design and develop an effective E-learning system that can satisfy individualized needs, a perfect software architecture is necessary.

II. SOFTWARE ARCHITECTURE

A. Definition

Software architecture includes one, or one group of, software components as well as the software components' external visible features and inter-relations [4]. Among them, externally visible features of the software refer to service, function, features, error processing, and information sharing, etc.

B. Function

Software architecture has a significant impact on the non-functional needs of the system; to be specific, the functions of constructing a good software architecture are

1) Customization of the system according to needs. It should encapsulate business logic into software components, select and assemble components freely according to needs, and build a dynamic customized application software system conveniently and flexibly.

2) Easy integration among system components. The restrictions of the architecture should regulate every components' interface and loosely couple components in the system without dependence on certain programming language. Then the plug-in and dynamic bind of the components can be easily realized. Meanwhile, integration of the other components and application system would become convenient with an open system interface.

3) High-levels of software reuse. Software architecture is reusable. It can be reused in many systems that have similar needs, which is more significant than code-level reuse. Through abstraction of the architecture, the designer can reuse some of the architecture that has been proved effective by practice and, thereby, improve the efficiency and reliability of the design.

C. Specific Domain Software Architecture and Its Construct Method.

Usually architecture specifically designed for certain software is not suitable for different domains and applications. To solve the problem, domain-specific software architecture appeared. It focuses on a certain domain (such as a telecommunication domain, a finance domain, etc.), abstract common features through domain analysis, or software architecture designed according to this specific domain and in this way effectively increases the reuse rate of software architecture and instantiates a series of reliable software products effectively and quickly. Therefore, compared to specified software architecture, it is helpful to have a deep understanding of the domain, the development of components oriented for the reuse and improvement of the software production rate to construct domain-specific software architecture [5].

a. Definition

Domain-specific software architecture consists of a domain model, reference requirements, and referential architecture used to support a group of applications in a specific problem domain, and its aim is to support the application in one specific domain [6].

b. Function

1) Software development in certain domains is more feasible. System development is based on indirectly mapping a problem space to a solutions space in software architecture through description of abstract layers of the system structure. However, it is difficult to implement due to the complexity of the problem space, abstraction and implementation. On the contrary, the development method of the domain-specific software architecture first divides the problem space into different domains and then realizes the solving scheme in this domain. Practice has proved this kind of software development more feasible [7].

2) Software reuse is easier in specific domains. Through deep analysis of specific domains and abstraction of the common features and dynamic behavior of the application systems in the domain, it is possible to apply the architecture to other application system development in the same domain, to reuse the software architecture, and to reduce the complexity of architecture.

c. Construction method

Domain-specific software architecture is constructed after domain analysis and on the basis of the domain model [8]. To identify the domain model, we determined the scope of domain with the help of an expert in the domain development environment. Then we designed an integration scheme according to the identified domain model component. Finally, we constructed the domain architecture and a mapping domain model for a realizable information framework. The model is shown in Figure 1.

Because a single E-learning system can hardly satisfy individualized application in the teaching process and because it is difficult to integrate many systems, we put forward construction of domain-specific software architecture oriented to E-learning. Within the restrictions of the architecture, we encapsulated the public business logic in the domain into the software architecture and stipulated the standard component interface. Through the plug-in and dynamic binding of the components, we built an individualized E-learning system. In the meantime, the standardized and opened interface made reuse of the components and the system integration possible and built a flexible E-learning system to support individualized teaching.

There are three steps in building E-learning oriented software architecture: 1. build an E-learning oriented domain model; 2. analyze the related components as well as the building and integration according to the domain model; and 3. build the E-learning oriented software architecture on the basis of the second step.

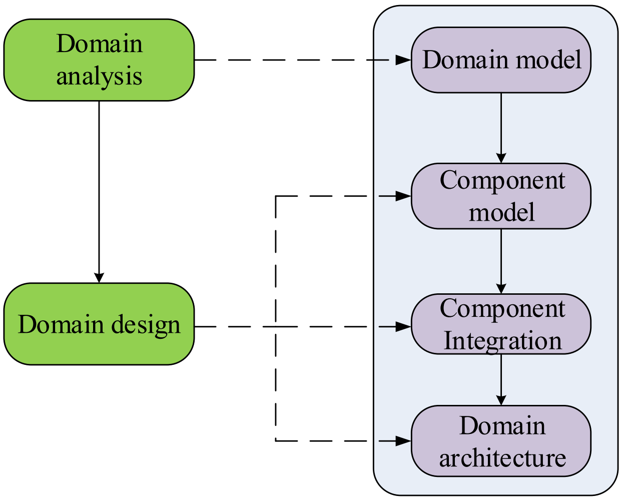


Figure 1. Construction model of domain-specific software architecture

III. SOFTWARE ARCHITECTURE ORIENTED TO E-LEARNING DOMAIN

A. Building of the Domain Model

The aim of developing E-learning oriented software is to effectively support all the activities and functions in the instructional mission. Therefore, all the informationized instructional models could be the guiding basis for the design and development of the learning environment, learning resources, and support tools for E-learning. An informationized instructional model is a technology-based instructional model or informationized learning model [9].

Professor Zhixian Zhong categorized the informationized instructional model into ten types: source based topic instructional model, program based instructional model, problem based instructional model, case study based instructional model, WebQuest instructional model, situational instructional model, internet cooperated study based instructional model, conceptual map based instructional model, electronic file based instructional model, and multi-intelligence based individualize instructional model[9].

Professor Kedong Li proposed that there are four learning models in digital learning: a ‘situation-explore’ model, a ‘resource using-topic explore-cooperative learning’ model, a ‘group cooperation-long-distance consulting’ model, and a ‘topic explore-internet development’ model [10].

Based on the instructional model, the learning model, and a detailed analysis of every model, we were able to abstract the common features of these models and determine the domain model of an E-learning domain.

In every model, there are three main targets: the learner; the learning content, including the learning target and the related learning resources; and the teacher, and they use the E-learning platform to interact. The interactions include those between the learner and the system, between the learners, between the learner and the teacher, and between the teacher and the system, etc. Therefore, this project built an E-learning domain model in regard to the interactions of the targets of E-learning, as shown in Figure 2.

The model has five levels: interactive object layer, interactive tools layer, interactive mode layer, interactive relationships between objects layer, and interaction layer, respectively from the inside to the outside.

a. Interactive object layer. The targets in the development of the E-learning system are the learner, the instructor, and the learning content.

b. Interactive tools layer. It needs related supports to make interactions between the learner, the teacher and the learning content. It mainly includes the related tools helping the learners to learn and the instructors to create instruction content and issue learning resources.

c. Interactive mode layer. It includes a synchronous mode and an asynchronous mode. It uses different interaction modes with related tools according to different needs: for example, synchronous audio, synchronous video, asynchronous audio, whiteboard, etc.

d. Interactive relationships between objects layer. This level involves six interaction relationships between the interaction targets: between the learner and learner, learner and instructor, learner and content, instructor and instructor, instructor and content, and content and content.

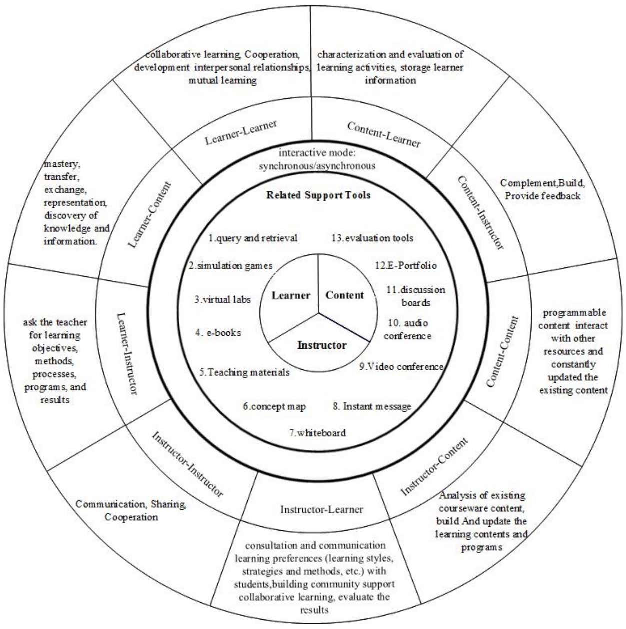

collaborative learning, Cooperation, development interpersonal relationships, mutual learning
characterization and evaluation of learning activities, storage learner information
Learner-Learner
Content-Learner
interactive mode: synchronous/asynchronous
Related Support Tools
mastery, transfer, exchange, representation, discovery of knowledge and information.
Learner-Content
Content-Instructor
Complement,Build, Provide feedback
1.query and retrieval
13.evaluation tools
2.simulation games
12.E-Portfolio
3.virtual labs
Learner
Content
11.discussion boards
4. e-books
10. audio conference
Instructor
5.Teaching materials
9.Video conference
6.concept map
8. Instant message
7.whiteboard
ask the teacher for learning objectives, methods, processes, programs, and results
Learner-Instructor
Content-Content
programmable content interact with other resources and constantly updated the existing content
Instructor-Instructor
Instructor-Content
Communication, Sharing, Cooperation
Instructor-Learner
Analysis of existing courseware content, build And update the learning contents and programs
consultation and communication learning preferences (learning styles, strategies and methods, etc.) with students,building community support collaborative learning, evaluate the results

Figure 2. E-learning model

e. Interaction layer. This level realizes the different interactions between targets according to the different interaction tools.

1) Learner to learner

Learners cannot only learn the online course material together, but they also can develop interpersonal relationships through interaction. Specified interaction activities mainly include cooperative learning, group discussion, sharing learning resources, sharing learning methods, etc.

2) Learner to instructor

In E-learning, the interaction between the learner and instructor can be supported through multiple kinds of tools, such as student comments, questioning, feedback, discussion, assignment creation, answering, etc.

3) Learning to content

The interaction between the learner and content has always been the most important part of traditional education. E-learning also supports such kinds of interaction, such as note taking, uploading learning resources, downloading learning resources, practicing, assignments and tests, etc.

4) Instructor to instructor

Instructors interact with other instructors through the community and create chances for special development—for example, group preparing of lessons and sharing of instructional resources and instructional methods.

5) Instructor to content

The interaction between the instructor and content focuses on the creation of the learning content and the building of learning activities. The instructors can consistently update the learning content and resources: specifically, post course materials and resources, build a learning community, design a teaching plan, and develop teaching materials, design discussion topics, and correct assignments.

6) Content to content

The interaction between content and content is a newly developed form of educational interaction. Programmed content could interact with other information sources to actively update its content. For instance, learning navigation can get data from the present learner model and update the related information for the learner.

B. Component Recognition and Component Modeling

A component is a reusable software unit, and component reuse can be divided into coarse-grained reuse and fine-grained reuse according to the reuse level. In an E-learning domain, a component is the common software unit supporting the instructional model and completing specified functions.

Modeling on an E-learning domain and from the perspective of target relations and software retention and management, an E-learning domain can be recognized as the following five coarse-grained components: learning content component, learner component, instructor component, interaction tool component, and public component. Every coarse-grained component includes a group of fine-grained components for completing certain functions. The learning content component includes fine-grained components like a content navigation component, a content management component, and a test component.

To manage fine-grained components effectively, this paper puts every component into its correspondent component library according to keywords. In every component library, we compose logical related components into a component package according to the function of the component, which is to form a three level component structure as component—component package—component library. The functions and categorizations of every component library are as follows:

The learning content component library provides the development and management of the learning content, to be specific, a learning content development component package, a learning content management component package, an assignment component package, a test component package, etc. The assignment component package includes an assignment model component, a topic choosing component, a topic management component, an objective question correcting component, a subjective question correcting component, a standard answer component, an assignment list component, an assignment score analysis component, an assignment making component, a checking excellent assignment component, an assignment management component, etc. (Detailed categorization is omitted because of space limitations.)

The learner component library provides the statistics and management of the learners' personal information: to be specific, a digital file component package, a learners' personal information management component package, a learning status diagnosing component package, etc.

The instructor component library provides management of the instructor-related information and the instructional environment's building and maintenance: to be specific, the instructor's personal information management component package and the education information management component package, etc.

The interaction tool component library provides the specific interaction component and supports the interactive activities between the targets: to be specific, synchronous interaction component packages and asynchronous interaction component packages. Every component package includes specified interactive components.

The public component library provides management and maintenance of the system: to be specific, the user authority component, the system maintenance component, the system management component, the external interface component, etc.

To help the computer recognize and deal with these components, we built a component logic model. A component logic model is the abstract description of the essential features of the component. It adopts all kinds of models, which are easy to understand and express, to describe complicated components. It helps the manager and the user understand the multiplexing mode of the component and is also convenient for integration and utilization.

a. The logic model of the component

The logic model of the component includes the component type, component realization, providing the interface and dependent interface, as shown in Figure 3.

1) Component type: It indicates what kind of problems it deals with and what interface functions it provides.

2) Component realization: It is the concrete realization of the component type. One component type can have several component realizations.

3) Providing interface: It refers to the interface provided for the external system.

4) Dependent interface: It refers to the external system interface, which is necessary when the component is processing.

b. Integration of component

Component integration means connecting the suitable revised component in the component library or connecting them to the elements in the present developing program and forming new target software.

At present, the models of components are mainly CORBA、EJB and COM+. However, these three models are related to the specified realization technique, and every interior component of the model complies with a particular protocol, such as the transmission protocol of CORBA is IIOP, but the transmission protocol of EJB is RMI. Therefore, it is difficult to integrate different standard components all together.

To solve this problem, we proposed a component integration method based on WebService and an XML data bus. WebService is a system of standards, and it defines how applied programs realize the inter-operation on the Web. It is accessible as long as it fits the WebService standard. This method is flexible, reusable and able to do better encapsulation without depending on certain programming language.

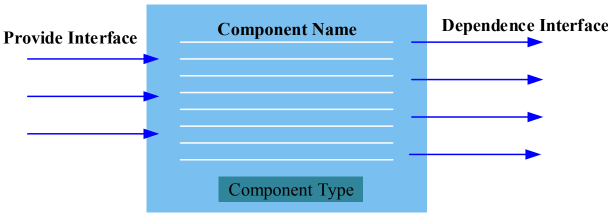


Figure 3. Component model

This method is mainly based on a UDDI/WSDL/SOAP protocol. First, it encapsulates all kinds of E-learning targets into reusable Web service components. Then it builds an XML data bus for coordination and transmission between components and realizes component plug-in and binds dynamically. The XML data bus is a pool of data entities and is compounded by XML and XML Schema. In this data pool, XML represents all the categories that are processing business data in the target model, and XML Schema is for building a target model from an XML structure for data persistence. Based on the XML data bus, every component completing business logic only depends on reading and writing without interacting with other components. Once the data bus is set, it would be simple to integrate different domain components and weave them into a black-box framework. It is dynamic and compatible and supports plug-in. Besides the features of WebService, it can be loosely coupled and does not depend on a specific language and operating system. Therefore, users can just focus on business logic and other domain problems in the program and do not need to care about the technical details in the developing process. Figure 4 shows the scheme of component integration and framework weaving based on an XML data bus.

C. *Modeling of E-learning Software Architecture*

For the entertaining and management parts of a certain business function or software architecture in the implementing process of concrete instructional model, it was only possible to satisfy the practical application of instruction so we assembled several components into a complete software architecture according to a concrete instructional model, that is, there are semantic relationships between components. Although we recognized all the components beforehand and managed them according to the category, the categorization did not completely describe the semantic relations between components. Because component granularity is too fine and a software developer who is not familiar with instruction would not know the business flow, it was difficult to form such fine-grained components into a complete E-learning system. Additionally, instructors are not good at developing software, so it was equally difficult to organically compose so many components according to instructional model.

To solve these problems, we used the reference from the framework in architecture domain and proposed the abstract framework oriented to the E-learning domain software architecture. An abstract framework is a class of components that can support an informationized instructional model/learning model, and it can integrate different components with different functions according to an instructional model/learning model. It can not only reflect the semantic relations between components, but its reuse level is superior to component reuse. Besides, as long as the external component satisfied the interface standards, it could add to its corresponding component library and connect to a certain abstract framework through the interface to achieve integration of different system components no matter what kind of technique it applied or its interior structure. In this way, we not only effectively reduced the complexity of the architecture, but we also made it convenient for instructors to choose a different framework and its corresponding components according to different instructional models and to custom-make a satisfactory E-learning system.

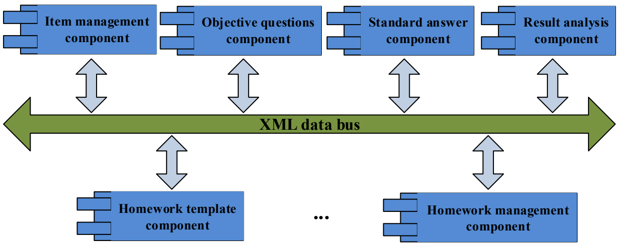

Item management component
Objective questions component
Standard answer component
Result analysis component
XML data bus
Homework template component
...
Homework management component

Figure 4. Component Integration Model

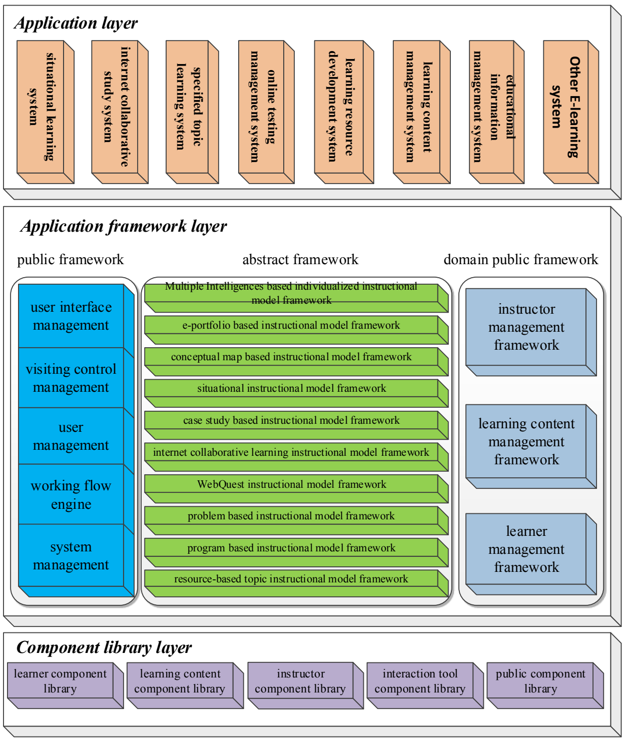

Application layer
situational learning system
internet collaborative study system
specified topic learning system
online testing management system
learning resource development system
learning content management system
educational information management system
Other E-learning system
Application framework layer
public framework
abstract framework
domain public framework
user interface management
visiting control management
user management
working flow engine
system management
Multiple Intelligences based individualized instructional model framework
e-portfolio based instructional model framework
conceptual map based instructional model framework
situational instructional model framework
case study based instructional model framework
internet collaborative learning instructional model framework
WebQuest instructional model framework
problem based instructional model framework
program based instructional model framework
resource-based topic instructional model framework
instructor management framework
learning content management framework
learner management framework
Component library layer
learner component library
learning content component library
instructor component library
interaction tool component library
public component library

Figure 5. E-learning domain software architecture

After introducing the domain abstract framework, we divided the E-learning domain software architecture into three levels, and they are component library layer, application framework layer, and application layer, respectively, from bottom to top. See Figure 5.

a. Component library layer

This layer is the basis of the whole E-learning software architecture. It is a group of reusable software units that had been tested in other projects. It mainly includes the learner component library, the learning content component library, the instructor component library, the interaction tool component library and the public component library. Every component library includes a component package or specified components. In addition, the component library could be extended. As long as the three part component complies with the rules of the component library layer, it can be added to the corresponding component library.

b. Application framework layer

The application framework layer is the gathering of all the frameworks. From the perspective of E-learning, we could integrate every component that supported the instructional model/learning model, form an abstract framework, integrate the related management component of the domain target, and form the domain public framework. From the perspective of the software, related components were needed to manage and maintain the software system, so we abstracted these components into a system public framework. Therefore, from the perspective of the E-learning domain and software management, the application framework layer is categorized into three types of framework: abstract framework, domain public framework, and system public framework, respectively.

To build an abstract framework is to integrate the related components of every instructional model concerned and to help the instructors adopt every instructional model in the actual instructing process. We combined the components of every core process of the instructional model and tools and solidified like models to form a corresponding abstract framework, such as a problem-based instructional model framework. In addition, since in actual teaching, teachers design different teaching activities according to different situational needs, we designed the part that satisfied certain instructional models and was easy to change into interface that allowed plug-ins to satisfy different instructional activities, such as whether the problem displaying media is text, picture, audio, video or multi-media. The abstract framework in Figure 5 was designed with reference to the informationized instructional model proposed by Professor Zhixian Zhong. It includes a resource-based topic instructional model framework, a program-based instructional model framework, a problem-based instructional model framework, a WebQuest instructional model framework, an internet collaborative learning instructional model framework, a case study-based instructional model framework, a situational instructional model framework, a conceptual map-based instructional model framework, an e-portfolio-based instructional model framework and a multiple-intelligences-based individualized instructional model framework. The domain public framework provides related management of the target in the E-learning domain and makes convenient extensions, including a learner management framework, an instructor management framework and a learning content management framework. The system public framework mainly provides the management and maintenance of the system, including a user interface management framework, a visiting control management framework, a user management framework, a working flow engine framework, a system management framework, etc. All of the products' architecture are interchangeable frameworks, but through the introduction of the application framework layer, we reduced the complexity of the software architecture and greatly increased the efficiency of reuse.

c. Application layer

This layer is a group of individualized and custom-made E-learning software architecture developed through choosing the framework and components, such as a situational learning system, an internet collaborative study system, a specified topic learning system, an online testing management system, a learning resource development system, a learning content management system, an educational information management system, etc. to satisfy the different needs for instructing and learning as well as learning management.

The instructors could adopt the following three modes to build an individualized, customer-made software system in the actual teaching process.

1) Choice of the abstract framework and related components. The instructor could choose the abstract framework directly from the application framework layer to satisfy the instructional demand, choose related components from the component library, and assemble them into a specified E-learning system.

2) Choosing component directly from the component library. The instructor could choose components to satisfy a corresponding instructional activity directly from the component library and assemble the E-learning system if no abstract framework could satisfy the actual instructional application in the application framework layer.

3) Extension of the abstract framework and component library. Besides through system extension, the instructor can add a satisfactory abstract framework that complies with the application framework layer standard into the application framework layer and add the components that comply with the component library layer standard into corresponding component library. In this way, we could satisfy more teaching and learning activities. In addition, in the processing of the system, instructors can add, omit or replace related components according to instructional needs at any time to meet the constantly changing instructional needs and achieve better instructional effects.

Through the flexible, layered and reusable E-learning domain oriented software architecture we proposed, we could meet the needs of different instructional models. By using this software architecture, instructors could custom make a system according to their instructional needs and achieve the goal of building an individualized E-learning system.

IV. CASE STUDY

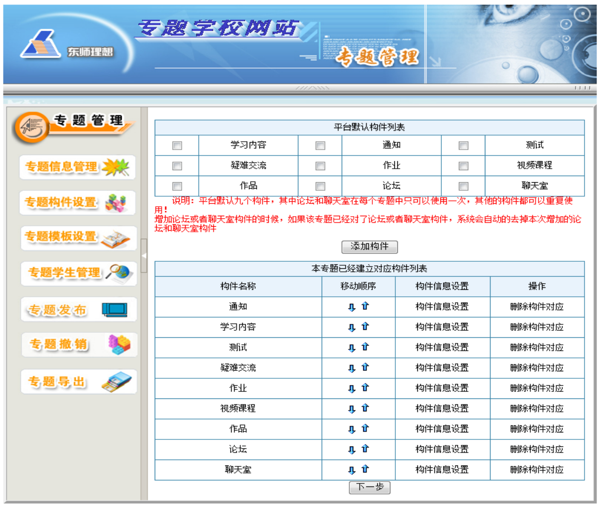
A. *Figures and Tables*

Based on the above E-learning software architecture, we developed a Web-based, flexible and customer-made topic learning website. This website could achieve resource-based topic instructional models.

What we called resource-based topic instruction refers the learners to a series of research activities about one topic that they can fully explore and utilize different resources, according to general rules and steps of scientific research, to improve their ability to problem-solve, explore, innovate, and promote their science and information capacity[9].

According to the traditional software development method, we needed to develop a software solely for resource-based topic instructional models or to integrate other software that had already achieved certain functions of the instructional model to meet the actual instructional needs. However, under the software architecture we proposed, instructors did not need to learn software design and programming. What they needed was just to choose a certain framework and related components, and they were then able to build a learning system to satisfy a resource-based topic instructional model.

In this case, the instructors could automatically come up with a learning system to satisfy specific needs by choosing a resource-based topic instructional model framework and a related public framework from the application framework layer. Then they chose satisfactory components from the component library, e.g. a system management component, and visited the control component from the public component library, choosing a user management component from the learner component library as well as a topic introduction component, test component, exercise component, learning content management component from the learning content component library. They chose an ask-answer component from the instructor component library as well as a room component, memo component, etc. from the interaction tool component library. This system covers all the functions, including learning content management, tests, exercises, ask-answer, memos, cooperation, user management, etc. Figure 6 shows the user interface of the topic learning website and Figure 7 is a page for building a Success English Corner topic learning website.



平台默认构件列表

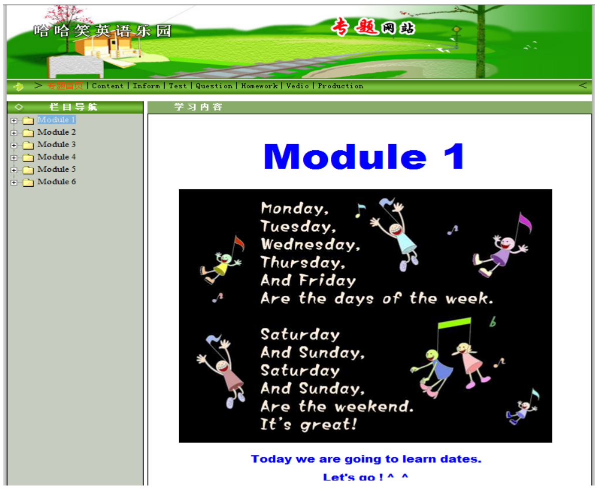
□	学习内容	□	通知	□	测试
□	疑难交流	□	作业	□	视频课程
□	作品	□	论坛	□	聊天室

说明：平台默认九个构件，其中论坛和聊天室在每个专题中只可以使用一次，其他的构件都可以重复使用！
增加论坛或者聊天室构件的时候，如果该专题已经对了论坛或者聊天室构件，系统会自动的去掉本次增加的论坛和聊天室构件

本专题已经建立对应构件列表

构件名称	移动顺序	构件信息设置	操作
通知	⇩ ⇧	构件信息设置	删除构件对应
学习内容	⇩ ⇧	构件信息设置	删除构件对应
测试	⇩ ⇧	构件信息设置	删除构件对应
疑难交流	⇩ ⇧	构件信息设置	删除构件对应
作业	⇩ ⇧	构件信息设置	删除构件对应
视频课程	⇩ ⇧	构件信息设置	删除构件对应
作品	⇩ ⇧	构件信息设置	删除构件对应
论坛	⇩ ⇧	构件信息设置	删除构件对应
聊天室	⇩ ⇧	构件信息设置	删除构件对应

Figure 6. The user interface of the topic learning website



Module 1

Monday,
Tuesday,
Wednesday,
Thursday,
And Friday
Are the days of the week.

Saturday
And Sunday,
Saturday
And Sunday,
Are the weekend.
It's great!

Today we are going to learn dates.

Let's go ! ^ ^

Figure 7. The content display page of the topic learning website

V. CONCLUSION

In this project, we built an E-learning domain model and an E-learning component model first and then proposed WebService and an XML data bus mode based component integration method. Then based on the informationized instructional model, we proposed an E-learning domain abstract framework that could express the semantic relationships among the components and achieve a higher level of reuse. On this basis, we finally formed a layered, E-learning oriented software architecture, which has a component library layer, an application framework layer, and an application layer. This architecture can layer multiplexing and does not depend on developing language and tools. Based on this architecture, instructors can flexibly build an individualized E-learning system, like building bricks, through choosing a framework and assembling and replacing components to meet the individualized needs of instructors and learners. Research shows positive results for this software architecture.

Further work would be on the diversity of the component library, effective organization and management of the component library, and research on component retrieval algorithms. We will introduce MDA thought, automatic generation of an abstract framework and support for a more informationized instructional model. In addition, we should do further investigation on the software evolutionary approach of E-learning software architecture.

REFERENCES

[1] Nichols, M. E-learning in context. E-Primer Series. 2008, pp 2.

[2] Ertmer, P. A.; & Newby, T.J. Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective. Performance Improvement Quarterly, 1993, 6(4), 50-70. http://dx.doi.org/10.1111/j.1937-8327.1993.tb00605.x

[3] Zhang, Y.S. Software Architecture. 2nd ed. Beijing: Tsinghua University Press, 2006.

[4] Bass, L.; Clements, P.; Kazman, R. Software Architecture in Practice(2nd ed). Boston: Addison Wesley Professional, 2003.

[5] Yang, F.Q.; Mei, H.; Li, K.Q. Software Reuse and Software Component Technology. Chinese Journal of Electronics, 1999, 27(2), 68-75.

[6] Tracz, W. DSSA (domain-specific software architecture) pedagogical example. Software Engineering Notes, 1995, 20(3), 49-62. http://dx.doi.org/10.1145/219308.219318

[7] Wang, X.H.; Guo, L.; Liu, Z.H. Research of DSSA-based ERP Modeling. Computer Engineering and Applications, 2006, 42(16), 224-232.

[8] Yu, Z.H.; Cai, Y.L. A Study on the Domain Specific Software Architecture. Microelectronics, 2004, 21(7): 66-69.

[9] Zhong, Z.X. Informatization Teaching Mode:Theory Construction and practice case. Beijing: Educational Science Publishing House.2005.12.

[10] Li, K.D. e-Learning: the core of information technology and curriculum integration. e-EDUCATION RESEARCH, 2001, 8,46-49.

AUTHORS

Peng Lu is with the Dept. of Media Technology and Communication, Northeast Dianli University, Jilin, China (peng.lu2008@gmail.com).

Xiao Cong is with the College of Science, Northeast Dianli University, Jilin, China (peng1984_2007@163.com).

Dongdai Zhou is with the Engineering & Research Center of E-learning, Northeast Normal University, Jilin, China (lup595@nenu.edu.cn).

This research was supported by the Northeast Dianli University Dr. Scientific Research Foundation Project (No. BSJXM-201219) and Scientific and technological research key program of Educational Commission of Jilin Province of China during the Twelfth Five-year Plan Period(No. 2015-239). Submitted 05 May 2015. Published as resubmitted by the authors 20 August 2015.