A Workflow based E-learning Architecture in Service Environment

KONG Weining, LUO Junzhou, ZHANG Tiantian
Department of Computer Science and Engineering, Southeast University
Nanjing, 210096, P. R. China
E-mail: {kweining, jluo, zhangtiantian}@seu.edu.cn

Abstract

Many e-learning systems integrate advanced network technologies like web services, grid computing to achieve learning at anywhere anytime. However, most of them have not taken account of the process of study activities; learners are difficult to navigate the learning resources adaptive to their own pace. Though several recent researches provide workflow-based e-learning, formal definition and analysis of learning process, organization of distributed learning objects still have not be fully considered. This paper first introduced the formal definition method of workflow-based courses using fuzzy Petri nets, and then proposed a workflow based e-learning architecture by utilizing workflow technology and learning object services. This architecture provides teachers tool for designing courses as composition of learning object services, and gives students the opportunities to achieve an orderly learning experience through adaptive navigating e-learning services step by step under the control of certain learning strategy. Some questions in application and future works are also discussed.

Keywords:
E-learning, web service, workflow, fuzzy Petri net

1. Introduction

E-learning gradually becomes the focus of people's interests as a new education style, which uses network technologies to break the temporal and spatial limit in traditional education with the hope to offer learners an individual learning experience at anywhere anytime. Web service technology are used to enhance the e-learning systems to communicate more efficiently and share data more easily [1][2], even grid computing begins to be considered in improving the computing capability of e-learning system [3]. E-learning systems have showed us their breaking the limit in flexibility, distributed computing and storage capabilities so as to provide an open and more efficient learning environment.

Nevertheless, learners are prone to reduce learning efficiency and even get lost during learning activities if they are lack of the strategy of searching and organizing learning resources in great amount of distributed resources. Some crucial tasks done in the traditional classroom have not been completely transferred into e-learning systems. We believe that a teacher's preparation for lessons is of great importance for his students. In the traditional teaching mode, teachers pay much time organizing the teaching contents and adjust them dynamically in the classroom so as to lead students to efficient learning. Some popular products, such as WebCT Vista [4], WBT TopClass [5], Lotus Learning Management System [6], still do not fully consider giving instructors opportunities to define a structural course for guiding students to navigate learning resources, although some of them allow assembling several different courses to build a new one. In recent years e-learning systems have promised to change the way people learn. However open issues still remain, in particular actual e-learning environments do not consider learning activities as part of the process of learning [7].

Workflow technology is recently used to design and develop process-centric courses and monitor student process, like Virtual Campus [7], Flex-eL [8]. Liu in [9] and Chen in [10] also provide Petri net-based formal method solutions. However, they all only integrate local learning resources; do not enable e-learning services. On the other hand, workflow itself lacks formal definition and analysis method.

Considering those disadvantages in process control of learning activities in e-learning systems, we proposed a workflow-based architecture in service environment which allows using extended fuzzy Petri net as formal method to define structural courses and to provide learners an adaptive learning environment for navigating learning resources one by one in sequence. By deploying the defined course in this architecture, different parts of a course provided by different services can be accessed regarding to learner's study progress.

This paper is structured as follows: In Section 2 we introduce the concept of workflow-based course and learning path, and the related technologies for describing and implementing it. Then Section 3 introduces the formal description of learning path. Section 4 shows the workflow-based e-learning architecture. Section 5 we discuss the algorithm and implementation issues on this architecture to achieve workflow courses. Section 6 illustrates how the learners can interact with our architecture through the learning path. Finally, Section 7 draws some conclusions and outlines directions for future work.

2. Workflow based course and learning path

Workflows are concerned with the automation of procedures where documents, information or tasks are passed between actors according to a defined set of rules to achieve an overall business goal [11]. Once a workflow is defined, it can be implemented by deploying in workflow management system (WFMS). Generally, in an e-learning environment where courses are defined as workflows, the participators of workflow are instructors and learners, the activities of workflow correspond to learning activities of learners such as searching learning objects, browsing learning objects, taking exams and so on.

In this paper, a learning path relates to the navigation sequence through a workflow course and a learning path node is a learning object associates with learning object service like a chapter or a section of a course. The activity of one node indicates invoking relevant LO service. There are also conditions and strategies for path choice between LOs, which can be added into operations like AND-SPLIT, AND-JOIN, OR-SPLIT, OR-JOIN, etc. Figure 1 is a simple learning path for a workflow course. LO B and C must be accomplished after A. LO D, E and F need to be finished after LO A, B and C. LO B and C are synchronous (AND-SPLIT/JOIN) which can be learnt at the same time with no requirements of sequence, while E and F could be chosen alternatively (OR-SPLIT/JOIN).

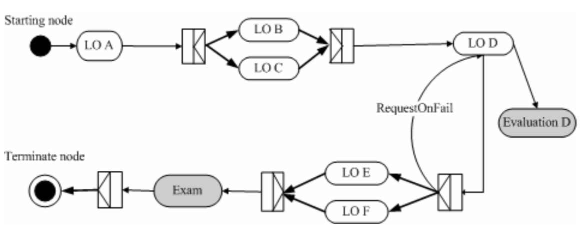

Figure 1 Simple Sample of Learning Path

We extend fuzzy Petri net for formal description of learning path. Looney [12] first modified the typical Petri nets which can represent the fuzzy production rules of a rule-based system to achieve fuzzy rule-based reasoning. In the e-learning system, a fuzzy Petri net is associated with a course and path definition with different learning content structures and different transition conditions according to students' learning result of the original place.

The implementation description of learning path takes the industry standard BPEL4WS (Business Process Execution Language for Web Services, BPEL for short) [13]. BPEL provides a language for defining workflows between web services which could fulfill different activities such as web service invoke, data manipulation, faults tolerance and process termination so as to combine them all together and build a complex process. These activities could be nested into structural activities defining the running mode of activities within. For example, serial or parallel are determined by certain conditions. BPEL calls service which interacts with it as partner which has a unique name and corresponding WSDL document [14]. Developers can define the relationships between partners through a set of rules. More important, the process can interact between users and services by starting with the "receive" operation to response the client request and ends with noticing users by "reply" operation.

The deployment and implement of learning path require workflow management system (WFMS). In WFMS, different instances of the same workflow could run simultaneously where each one has its own status. In e-learning environment, the same learning activities of different students can be run simultaneously and WFMS helps to maintain their own performance statuses. At present, the interaction between users and WFMS is base on documents. However, what kind of document should be submitted to WFMS by users when they complete studying a LO and want to know the next step is the problem we must solve. We will discuss it in Section 6.

3. Learning path description by fuzzy Petri net

After introducing the concept of learning path, we extend the fuzzy Petri net model [15] as E-learning net (ELN) to provide course definition with learning sequence. ELN can be graphically presented by two types of nodes which are places and transitions, where each place may or may not contain a token associated with a proposition and a truth value between 0 and 1, and each transition is associated with a certainty factory value (CF) and a threshold value between 0 and 1. Thus an ELN model can be defined as an 11-tuple:

ELN = $(P, T, D, I, O, f, \alpha, \beta, M, K_p, K_t)$, where

$P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of places;

$T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions;

$D = \{d_1, d_2, \ldots, d_n\}$ is a finite set of propositions;

$P \cap T \cap D = \Phi$, $|P| = |D|$;

I: $T \rightarrow P^{\infty}$ is the input function mapping from transitions to bags of places, a directed arc from p_j to t_i exists if $p_j \in I(t_i)$;

O: $T \rightarrow P^{\infty}$ is the output function mapping from transitions to bags of places, a directed arc from t_i to p_j exists if $p_j \in O(t_i)$;

f: $T \rightarrow [0,1]$ is an association function mapping from transitions to real values between zero and one, if $f(t_i)$ =

μ_i , where $\mu_i \in [0,1]$, t_i is said to be associated with a real value μ_i;

α: $P \rightarrow [0,1]$ is an association function mapping from places to real values between zero and one, if $\alpha(p_i) = y_i$, where $y_i \in [0,1]$ and $\beta(p_i) = d_i$, then the degree of truth of proposition d_i is y_i;

β: $P \rightarrow D$ is an association function, a bijective mapping from places to propositions, if $\beta(p_i) = d_i$, where $d_i \in D$, p_i is said to be associated with the proposition d_i;

M: $T \rightarrow \exp r$ specifies the addition modes for firing the transition;

K_p: $P \rightarrow E_p$ is one to one mapping from places to evaluation learning object set E_p;

K_t: $T \rightarrow S_p$ is one to one mapping from transitions to learning object set S_p.

Next we give detail explanation about some elements in ELN.

Places: denote learning object services which can be accessed from service providers, and the token in it maintains the status of the service. The learning object is associates with meta-data which describes the learning resources in a common way and facilities the search. There is also a kind of learning object we call it evaluation learning object which is developed according to IMS's Question & Test Interoperability Specification (QTI) [16] to help to evaluate the user's learning performance.

Transitions: move from one learning object to another. A transition t_i is enabled if for all $p_j \in I(t_i)$, $\alpha(p_i) \geq \lambda$, where λ is a threshold value and $\lambda \in [0,1]$. By setting different threshold values whether the user can pass current learning object can be decided. Firing transitions can be considered as firing fuzzy production rules. For example, 5 basic rule types according to [15] can be used to describe the learning path routing, the larger the value of μ_i, the more the rule is believed in:

1) Sequence R_i: IF d_j THEN d_k (CF $=\mu_i$)
2) AND-SPLIT R_i: IF d_j THEN d_{k1} and d_{k2} and … and d_{kn} (CF $=\mu_i$)
3) AND-JOIN R_i: IF d_{j1} and d_{j2} and … and d_{jn} THEN d_k (CF $=\mu_i$)
4) OR-SPLIT R_i: IF d_j THEN d_{k1} or d_{k2} or … or d_{kn} (CF $=\mu_i$)
5) OR-JOIN R_i: IF d_{j1} or d_{j2} or … or d_{jn} THEN d_k (CF $=\mu_i$)

M is a firing mode function: specifies the method how to trigger the firing of a transition. It could be:

Auto: the firing task can be triggered immediately when it is enabled without student's request.

Action: the firing task is triggered by student's actions like clicking a hyperlink on the web when it is enabled.

Delay: the firing task is triggered after a predefined period which can be used to force the student to compete his learning step within a certain time.

To see whether the firing condition is met needs tests for given LOs, because the evaluation results can be used to decide whether a student can move from one LO to the post-condition LOs. According to this consideration, we propose a evaluation learning object set E_p to which different LOs can be mapped by K_p. The Evaluation LO includes reusable evaluation information and interfaces for receiving user input and score output. In addition, each transition can be mapped to LOs set S_p by K_t to specify review LO if a student failed a test.

LOs and evaluation LOs have reusability to be used for different courses due to their standard metadata description. Different evaluation LOs can be used for the same LO, but only one evaluation LO can be bound to a LO at the same time. Thus, by setting different threshold values of a transition, even though using the same evaluation LO, the teacher can achieve different difficulty levels for a variety of students. Then comparing the token value as evaluation grade with the threshold value as passing level can see whether a student achieved the requirement. Here we present a simple evaluation procedure for calculating the token value: let the full grade of the evaluation be 1, an evaluation LO includes n evaluation items, their difficulties are $d_1, d_2, \ldots, d_n$ ($d_n \in [0,1]$), higher the value is, more difficult the item is, a student's actual grade for each item is m_1, $m_2, \ldots$, m_n ($m_n \in [0,1]$),

$$m_i = \begin{cases} d_i - \dfrac{\sum_{n} d_j}{n}, & \text{if wrong} \\ 1, & \text{if correct} \end{cases}$$

then the final score can be assigned to token value: $\frac{\sum m_i}{n} \rightarrow y_i$, y_i is the study grade, where $y_i \in [0,1]$. The higher the study grade y_i is, the more the evaluation result of certain part in a course is satisfied.

Let PR be a set of fuzzy production rules, NPR be a set of fuzzy production rules for normal navigation through the learning path, RPR be a set of fuzzy production rules for reviewing after failure on tests. The ith fuzzy production rule is formulated as follows:

IF $y_j \geq \lambda$, THEN $PR_i = NPR_i$ (threshold value = λ)
NPR_i : IF d_j THEN d_k (CF $=\mu_i$)
ELSE $PR_i = RPR_i$
RPR_i : IF d_j THEN d_l (CF $=\mu_i$, $d_l = R_p(t_i)$)

where y_j is the token value in d_j , its value is between zero and one and will be regarded as learning result from the student. The threshold value of t_i is λ which is used to determine the next reachable place according to using NPR_i or RPR_i. For example, d_j is "LO_j is accessed for learning", d_k is "LO_k is accessed for learning", d_l is "LO_l is accessed for reviewing", depending on the token value d_j and threshold value λ, transition t_i can decide whether fire to d_k for further learning on satisfied test result or to d_l for reviewing on test failure.

4. Workflow-based e-learning architecture

A defined learning path itself cannot run and interact with the learners, so there is need of an open architecture both for teachers to define, map to composition services, deploy their courses, and for learners to be guided by that architecture.

An open and self-centered learning environment not only shares learning objects, but also provides enactment of learning path for various roles. The emerged web service and grid service technology bring us an open environment where learning objects can be searched in registry and accessed through portTypes. Workflow technology can combine services providing different learning objects and implement orderly learning activities by invoking relevant services.

Therefore, we attempted to integrate web service and workflow technology to build a workflow-based learning architecture WELA. WELA contains the WFMS, a web server, a collection of Web Services, some repositories and other useful tools. Figure 2 demonstrates the main architecture of WELA.

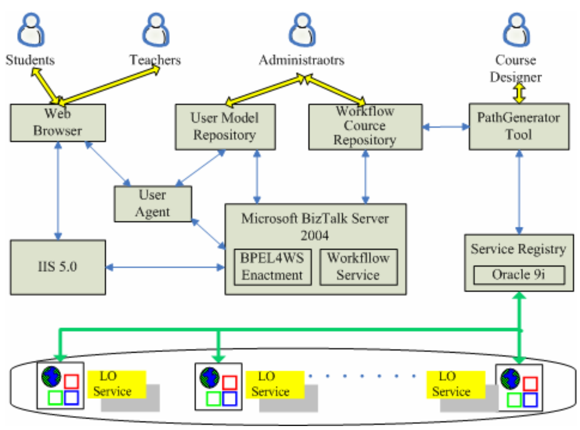

Figure 2 Workflow based e-learning architecture

LO Service There are many standards in description of learning object metadata. This architecture is as a test bed for CELTS (Chinese E-learning Technology Standard, http://www.celtsc.edu.cn). CELTS standard specifies a data schema for describing the structure of a metadata instance. So we could use CELTS schema to describe the attributes of learning object and the binding language of CELTS schema is XML. The use of CELTS for description of metadata could greatly facilitate searching of learning objects.

As we can see, there are various LO services at the lowest layer of WELA, which are managed by various learning objects providers to provide different learning resources in different areas. These services could be distributed physically and run on different platforms. They also could be web services or grid services (Recent system mainly uses web service. We will offer support for grid service in later research). Each LO service has individual hardware and software environments and organization strategies.

LO services provide functions as follows:

To publish learning objects by uploading metadata XML files to LO service registry;

To manage published learning objects including update and delete operations;

To support searching of learning objects in local area based on metadata;

To support acquisition of learning objects;

LO service registry LO services publish respective WSDL documents in LO service registry. They also publish learning objects by uploading metadata XML files of learning objects. Meantime, service registry supports searching of registered learning objects with different searching conditions/in different searching ways. Oracle 9i database which supports XML files storage and search is used in this registry.

PathGenerator Tool PathGenerator is a tool to define courses with learning path. Course designers (When there isn't such special role, it could be competent manager or teacher.) search for the needed learning objects by PathGenerator from LO service registry, define the relationships and orders of them, add in strategy of path choice, and finally establish a workflow course template described by BPEL which will then be transferred to workflow course repository. PathGenerator also generates corresponding VML (Vector Markup Language) code [17] for each template to facilitate visual display in web browsers. Course designers can also establish different templates for a single course so as to implement different learning targets for different learners.

Workflow Course Repository When a student starts a workflow course, WFMS takes the course template chosen by learners or designated by instructors from workflow course repository and generates an instance of such template. Workflow course repository allows course designers to view course templates available and modify on recently course templates to create a new one while it also permits administrators or instructors to view them.

User Model Repository A well-designed user model should include a user's knowledge and past learning experiences as an important evidence for system to provide individual education, so that the system can skip the knowledge a student already mastered to help he/she finds the information actually needed quickly. In WELA, user model is designed as a copy of workflow course with additional status value in each activity node to track learner's study paces. When an instance of a workflow course restarts, WFMS bypasses completed nodes according to user model and fast locates the node the learner needs currently. To users, it seems not to start from the first node. (Actually, workflow instance still begins from the first node but just jumped over some unnecessary nodes).

WFMS We choose Microsoft Biztalk Server 2004 as the workflow engine in WELA. A course with defined learning path actually determines the process of how and when to get each learning object. WFMS has to follow the course definition to call relevant services. Course learning includes the request to LO service and the interaction with students which are implemented in BizTalk by invoking external services and local service. External services correspond to the LO services while local service interacts between learners and WFMS.

5. Generating learning path

Generations of learning path for a course contains two important steps which are search for needed LOs from description of published LOs metadata, mapping the ELN to combination of needed LO services and publishing them into workflow management system. Now we introduce the design and implementation issues from two aspects.

5.1 Search of learning objects

First, we need to find out the relevant learning objects according to description of all learning objects metadata before defined a workflow-based course. In the platform adopts WELA, we need a method to search specific information included in metadata XML in service registry. We choose XQuery[18] as the search method for learning objects because XQuery is the standard designed by W3C for searching several kinds of XML source formats. It is of great importance that it could carry out search with complex condition not only in a single XML file but also among several XML files, rational databases, objects and other non-structural documents. This property does accord with the requirements of cross-searching for relevant LOs in LO service registry.

5.2 Mapping ELN to services composition

Next, we generate the implementation description of learning path defined by PathGenerator Tool. The idea of choosing BPEL comes from that course is an integration of several learning objects and each learning object is provided by learning object service. As a standard process description language, BPEL does not rely on detail workflow management system which facilitates easy system update or replacement. Thus, BPEL is exactly the suitable standard which integrates the services on Internet in our architecture. PathGenerator gets the WSDL documents of services where each LO came from, and then takes defined ELN and corresponding WSDL documents as the evidence for the BPEL code. Due to the support of structural activities by BPEL, activities which are to be run could be described as orderly structure. Therefore, fuzzy production rules could be translated into activities in BPEL. Tab 1 gives the correspondence between them.

Fuzzy production rule	BPEL description	BPEL Function
Single rule	Sequence	Including a serial of orderly run activities and a final activity to end process
“or” rule	Switch	Choosing a branch with *true* condition
“and" rule	Flow	Synchronization and parallel between activities

Tab 1 Correspondence between fuzzy production rules and BPEL

Suppose that we have already had a course defined by ELN model, the works that should be done in advance includes:

1) Define each part of a course. Normal propositions d_1, d_2, ..., d_n are drawn from the main part of a course, the reviewing propositions d_{n+1}, d_{n+2}, ..., d_{n+m}, where $n \geq m$.

2) Define the mapping function from LOs to evaluation LOs wherever the LO needs an evaluation procedure.

3) Define the production rules. By setting the threshold value λ to each transition, the relationship between normal propositions and reviewing propositions will be established.

Figure 3 shows a defined ELN according to Figure 1.

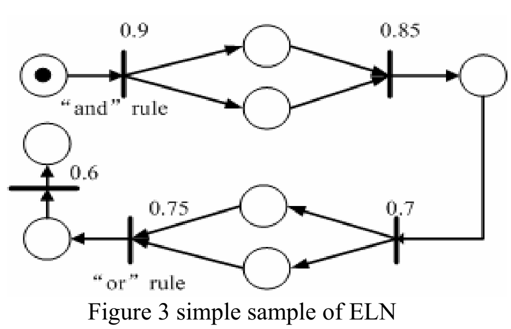

Figure 3 simple sample of ELN

Now, we propose the brief mapping algorithm as follows, here, a concept from [15] should be introduced, the set of places which is immediately reachable from a place pi is called the immediate reachable set of p_i and is denoted by IRS(p_i).

0-7695-2432-X/05 $20.00 © 2005 IEEE

Let PlaceActivity(p) be the function which generates codes for invoking the external LO service that provide LO according to p and the associated evaluation LO:

if $K_p(p) \neq \Phi$, the token in p has a mapping evaluation LO **then**

<sequence>
<invoke> LO service for p </invoke>
<flow>
<invoke> LO service for $K_p(p)$ </invoke>
<invoke> User Agent </invoke>
</flow>
</invoke>
</sequence>

else *<invoke> LO service for p </invoke>*

where User Agent service is an internal service, through its parallel execution with another external LO service, it can interact with the students to receive their input.

Step 1:

Let p_r be the starting place, generate a BPEL process, add <process> element;

let parent place $p = p_r$, where p is the current node to which the BPEL activities will be add

Step 2:

ELN2BP(p) is a recursive function for mapping places, production rules and transitions into BPEL process:

Let t_p be transition whose input place is p , $p \in I(t_p)$;

let PR_t be the production rule mapped to t_p;

First, mapping the firing mode function M to BPEL code before firing a transition between p and IRS(p):

if G(t_p) = auto **then** generate active by PlaceActivity(p);

else if G(t_p) = action **then** generate <flow> activity for PlaceActivity(p) and <invoke> activity which calls the internal User Agent service to receive client's action request;

else if G(t_p) = delay **then** generate <wait> activity which will remain idle for a given time period utile firing the transition.

Next, we map different production rules to composition services.

if PR_t contains "and" and $|O(t_p)| > |I(t_p)|$, transition t_p is according to AND-SPLIT operation, **then**

generate <switch> activity, in case of input token value $\alpha(p)$ is more than threshold value, generate <flow> activity, for each $p_i \in$ IRS(p), insert active by PlaceActivity(p_i); in case of $\alpha(p)$ < threshold value, PlaceActivity($K_t(t_p)$) for firing reviewing production rule.

else if PR_t contains "or " and $| O(t_p)| > | I(t_p)|$, transition t_p is according to OR-SPLIT operation, **then**

generate <switch> activity, in case of input token value is more than threshold value, generate<switch> activity, each case is implemented by PlaceActivity(p_i) and case condition is $\alpha(p) - \lambda$, where $p_i \in$ IRS(p), $t_p \rightarrow \lambda$, in case of $\alpha(p)$ < threshold value, PlaceActivity($K_t(t_p)$).

else if $|O(t_p)| = |I(t_p)| = 1$, it's a sequence operation **then**

generate <switch> activity, in case of $\alpha(p)$ > threshold value, insert active by PlaceActivity(p_i), where $p_i \in$ IRS(p), in case of $\alpha(p)$ < threshold value, insert activity by PlaceActivity($K_t(t_p)$).

else if $|O(t_p)| = 1$ and $|I(t_p)| \neq 1$, t_p is either AND-JOIN or OR-JOIN transition, **then**

the same procedure as above sequence operation.

Last, for each $p_i \in$ IRS(p), let $p = p_i$, if IRS(p) $\neq \Phi$, call ELN2BP(p), else p_i is either a leaf node or terminate node, return.

Step 3:

In a workflow-based course, the end of the course should come a definite terminate node. So if p is a leaf node, throw an error information to remind the teacher to redefine the ELN model, and if p is the terminate node.

6. Interaction between user client and workflow-based e-learning architecture

Workflow management system as a backbone of WELA plays an important role as a bridge between students and LO services, but BizTalk is designed for interaction with web services or code plug-ins. However, in a fruitful learning environment, WFMS needs to track students' activities [7]. So, we have built an interaction mechanism based on a designed local service we call it User Agent service. When the WFMS needs to get the user's input, it calls the User Agent service; the User Agent service waits for user's input and transfers it to WFMS. User Agent service is not only a service called by WFMS in WELA, but also responsible for maintenance of user models during the workflow run-time.

As a course template has user models according to it, which give important information for tracking study status and designing personalized course template, a mechanism for tracking user's activities in course study is needed. It is obvious that whether a user has went through a node in learning path can be decided by checking whether he has accessed the LO related to that node. If a node in learning path doesn't bind any evaluation LO, an access to the relevant LO can be simply considered as having experienced the node. As far as considering a node binding an Evaluation LO, we use six node statuses as following: passed, completed, browsed, incomplete, failed, not attempted. The initiated status of a node is set to "not attempted", and before evaluation procedure of a LO, the node status can be set to browsed, completed or uncompleted, but as soon as the node status is assigned "completed", the evaluation LO will be called.

When a user starts a workflow course instance, a local User Agent service will be called at the same time a node active starts, User Agent service gets all node status information from the user model, bypasses all "passed" nodes and locates the node the user currently needs to learn. During the runtime of a workflow course, User Agent listens to user client's input, calls the corresponding next node activity, and returns the LO to the user. Figure 4 shows the GUI presented to the user. On the right is the graphic presentation of a workflow course using VML code, and Microsoft Wizard has been chosen to present a vivid User Agent image service to the user.

Figure 4 GUI of Learning Path and User Agent

7. Conclusions and Future Work

In this paper, a well-structured e-learning architecture supported by the power of WFMS has been designed. A main contribution of this research is the formal description of learning path and the application of workflow technology in building and implementing workflow course in service environment. Our approach showed that students can benefit from this approach to study on their own pace, with own time arrangement by the guide of learning path.

The following research items need further discussion:

Design more rules for learning path choosing, support different composition ways for building larger course using existing course templates.

In a typical learning path, retrieving a learning object is achieved by calling a LO service. If the LO service becomes a grid service, a solution should be given to the grid services composition with BPEL.

8. Acknowledgement

This work is supported by National Natural Science Foundation of China under Grants No. 90412014 and the Key Scientific and Technological Program for the Tenth Five-Year Plan of China under Grants No. 2001BA101A12.

9. References

[1] G. Vossen, Peter Westerkamp, E-learning as a Web Service, *Proceedings of the Seventh International Database Engineering and Symposium*, IEEE 2003

[2] Zhengfang Xu, Zheng Yin, and Abdulmotaleb El Saddik, A Web Services Oriented Framework for Dynamic E-Learning Systems, *CCECE 2003 – CCGEI 2003*, Montreal, May 2003, pp 943-946

[3] V. Pankratius, G. Vossen, Towards E-Learning Grids: Using Grid Computing in Electronic Learning, Proc. *IEEE Workshop on Knowledge Grid and Grid Intelligence*, Saint Mary's University Halifax, Nova Scotia, Canada.

[4] WebCT. Vista. http://www.webct.com.

[5] WBT. Topclass e-learning suite. http://www.wbtsystems.com.

[6] IBM Lotus learning management system. http://www.lotus.com/lotus/offering3.nfs.

[7] Nicosia, Carrying on the e-learning process with a workflow management engine, Symposium on Applied Computing, *Proceedings of the 2004 ACM symposium on Applied computing*, 2004.

[8] Lin J., Ho C., Sdiq W., Orlowska M. E. (2001). On Workflow Enabled e-learning Services. *IEEE Advace Larning Technology Conference*, pp 345-352

[9] Xiao-Qiang Liu, Min Wu, Jia-Xun Chen, Knowledge aggregation and navigation high-level Petri nets-based in e-learing, *Proceedings of the First International Conference on Machine Learning and Cybernetics*, Bejing, Nov, 2002

[10] Juei-Nan Chen, Yueh-Min Huang, Using dynamic fuzzy Petri net for navigated learning, *Exploring Innovation in Education and Research* , Tainan, Taiwan, Mar, 2005

[11] Workflow management coalition document wfmc-tc-1011.

[12] C.C. Looney, "Fuzzy Petri nets for rule-based decision making", *IEEE Transaction on Systems, Man, and Cybernetics*, Vol 18, No.1, 1988, pp 178-183.

[13] Business Process Execution Language for Web Services Version 1.1. http://www-900.ibm.com/developerWorks/cn/webservices/ws-bpel_spec/index.shtml.

[14] Web Serices Description Language (WSDL). http://www.w3.org/2002/ws/desc/.

[15] S. M. Chen, J.S. Ke, J.F. Chang, "Knowledge representation using fuzzy petri nets", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 2, No. 3, 1990, pp. 311-319.

[16] IMS Global Learning Consortium, Question & Test Interoperability: ASI Information Model Specification Version 1.2, February 2002.

[17] Vector Markup Language, http://www.w3.org/TR/1998/NOTE-VML-19980513

[18] "XQuery: A Query Language for XML", Februay 2001, W3C Working Draft