A multiagent and service-oriented architecture for developing adaptive e-learning systems

Fuhua Lin,* Peter Holt, Steve Leung and Qin Li

School of Computing and Information Systems,
1 University Drive, Athabasca,
Alberta, Canada T9S 3A3
E-mail: oscarl@athabascau.ca
E-mail: holt@athabascau.ca
E-mail: stevel@athabascau.ca
E-mail: qinl@athabascau.ca
*Corresponding author

Abstract: This paper presents an architecture for developing adaptive e-learning systems using intelligent agent technology and Web Services (WS) technology. Intelligent software agents are designed for supporting users to accomplish knowledge-intensive tasks. WS are designed for the integration of distributed knowledge and information resources and exposed as standard services using widely accepted protocols. In particular, the Domain Model Ontology in the architecture can be shared between applications in the Semantic Web setting for reuse and interoperability. To demonstrate the feasibility of the proposed approach, we implement a prototype of a programme planning and scheduling system for adaptive e-learning.

Keywords: intelligent agents; service-oriented architecture; adaptive e-learning systems; ontology; Semantic Web.

Reference to this paper should be made as follows: Lin, F., Holt, P., Leung, S. and Li, Q. (2006) 'A multiagent and service-oriented architecture for developing adaptive e-learning systems', *Int. J. Continuing Engineering Education and Lifelong Learning*, Vol. 16, Nos. 1/2, pp.77–91.

Biographical notes: Dr. Fuhua Lin is an Associate Professor in the School of Computing and Information Systems at Athabasca University, Canada. He is conducting research in Intelligent Systems, Multiple Agent Systems and Knowledge Engineering and has published about 40 papers in journals and refereed conference proceedings in these areas. He has worked as an Assistant Research Officer in the Institute for Information Technology (IIT) at the National Research Council (NRC) of Canada and as a postdoctoral fellow in Intelligent Systems Group at the University of Calgary, Canada. He received his PhD in Industrial Engineering and Engineering Management from the Hong Kong University of Science and Technology (HKUST) in 1998, his MSc in Artificial Intelligence in 1992 and BSc in Mathematics in 1982 from the South China University of Technology, China.

Dr. Peter Holt is a Professor in the School of Computing and Information Systems at Athabasca University, Canada. He is conducting research in the human–computer interaction, applications of AI, the application of computers

and other technology in distance education, models of human cognition, planning computer systems, research design, applied statistics, development of learning materials, management models for distance education and managing technological change. He received his PhD in Cognitive Psychology from the University of Alberta, Canada in 1982.

Steve Leung graduated in Sociology in 1987 and received his MSc in Information Technology in 1989. After graduation he worked mostly on project management capacity to develop and implement software solutions for private companies and multinational corporations. Since 2001 he has been working as a part time tutor at the Athabasca University and at the same time participating in some research projects. Currently, he is working as an Academic Coordinator for Athabasca University, responsible for the delivery, maintenance and development of courses in Java Programming, Distributed Systems and Human Computer Interaction.

Qin Li is an MSc student in the Department of Electrical and Computer Engineering at the University of Alberta, Canada. He is working as a research assistant in the School of Computing and Information Systems at Athabasca University. His research interests include software engineering and web services.

1 Introduction

Adaptive and collaborative e-learning systems can potentially deliver personalised electronic course material and services and are therefore able to accommodate a large variety of learners. Developing such systems, however, are typically complex because they involve many dynamically interacting components, each with its own need for knowledge and information resources and the complexity of coordinating the components.

An adaptive e-learning system should be scaleable and easy to extend. A small change in the domain knowledge should not require an intensive system-wide modification to alter the information and all the functions that initiate actions are based on this changing information.

The agent-based approach is suitable for supporting adaptive e-learning since relationships among learners, courses and instructors have been lasting for a considerable period of time (Chan, 1995). An intelligent agent is a computer programme that is situated in some environment, which is capable of autonomous action and learning in its environment to meet its design objectives. More research has shown that an educational environment can be enhanced by a set of software agents by reducing information workload and for automatically performing many knowledge-intensive tasks for both learners and educators (Baylor, 1999; Greer et al., 2001; Thaiupathump et al., 1999). However, people have faced many challenges in developing this technology for commercial applications, mainly because of the lack of an accepted industry standard method for the development and implementation of agents and 'agent-ready' environments where agents can live and run.

A web service is a software system identified by a Universal Resource Identifier (URI), whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the web service in a manner prescribed by its definition, using

XML-based messages conveyed by the internet protocols (W3C Web Services Architecture Working Group, 2002). Web Services (WS) technology is characterised by a standardised communication protocol, interoperability, easy integration and development. It provides an excellent architecture for developing service-based learning systems. However, WS have some limitations (Huhns, 2002). For instance, they are passive until invoked. We believe that overcoming these limitations require the integration of agents and WS.

In this paper, we propose an architecture in which WS are used for integrating and exchanging distributed knowledge and information resources and as complimentary partners with intelligent agents supporting adaptive e-learning. In particular, the Domain Model Ontology in the architecture can be shared between applications in the Semantic Web setting and used for the purpose of information exchanging.

2 Related work

Dicheva and Aroyo (2004) proposed a general reference architecture for supporting sharing and exchanging information between adaptive concept-based Web Educational Systems. Brusilovsky and Nijhawan (2002) proposed a framework for adaptive e-learning based on distributed reusable learning activities. Mitrovic and Devedzic (2004) proposed the M-OBLIGE model for building multitutor ontology-based learning environments. The model allows domain expertise to be shared and can be used as a framework for integrating multiple tutors on the web. Moreale and Vargas-Vera (2004) developed an e-learning services architecture offering semantic-based services to students and tutors, in particular, ways to browse and obtain information through WS. The learning services architecture and learning services stack have been proposed and developed by the Learning Systems Architecture Lab at Carnegie Mellon University (Blackmon and Rehak, 2003). The main aspect of adaptive web-based educational systems in the context of achieving interoperability in the Semantic Web has been summarised by Aroyo and Dicheva (2004).

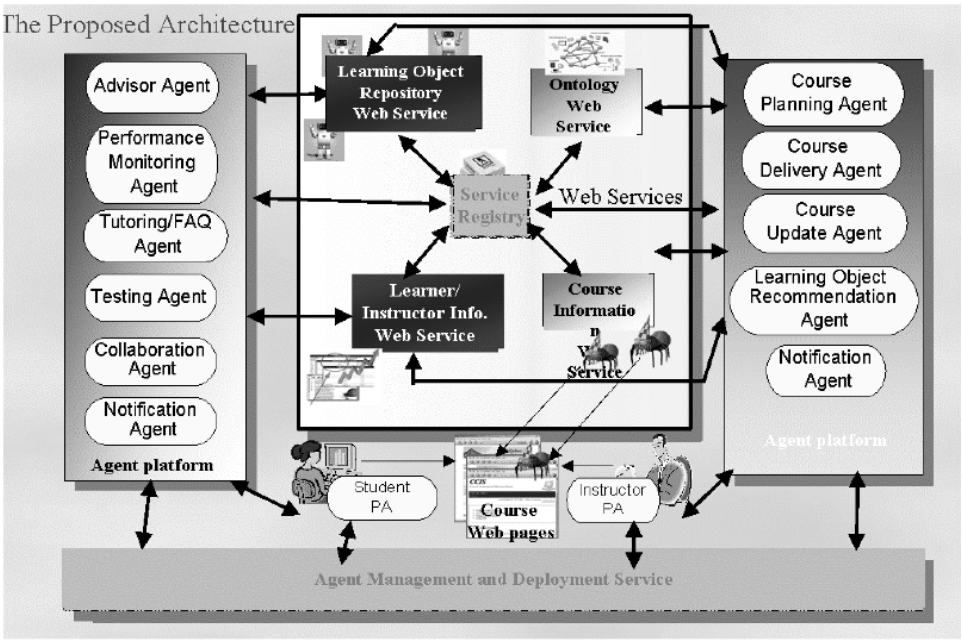
Recently, the term ontology started to spread and became more commonly used. In our research on adaptive e-learning, an ontology is a formal explicit description of concepts in a domain of discourse (classes or called concepts), properties of each concept describing various features and attributes of the concept (slots, called roles or properties) and restrictions on slots (facets or called role restrictions). The combination of an ontology and a set of instances of its classes make up a knowledge base. Kay (1999) and Chen and Mizoguchi (1999) noted the advantage of using ontologies for learner/user models. Mizoguchi and Bourdeau (2000) studied how ontologies can help to overcome problems in artificial intelligence in education. Razmerita et al. (2003) proposed a generic ontology-based user modelling architecture, OntobUM, based on the IMS LIP (IMS LIP, 2005).

The Ontology Web Language (OWL) (2004) proposed by W3C is used to publish and share ontologies, supporting advanced web search, software agents and Knowledge Management. Protégé 2000 is a knowledge-based systems editor and browser, which allows the domain experts to build knowledge-based systems by creating and modifying reusable ontologies and problem-solving methods. It has recently been adapted to work with OWL files. The authors used Protégé 2000 (http://protege.stanford.edu/) to model Domain Ontology for e-learning (Hogeboom and Lin, 2004).

3 The proposed architecture

In the proposed architecture, an adaptive and collaborative e-learning system consists of users' personal agents, task agents running on distributed agent platforms and a set of discoverable WS (see Figure 1).

Figure 1 The proposed architecture



3.1 Personal agents

A Personal Agent (PA) or User Interface Agent (UIA) is a GUI-driven interface between a user and an e-learning environment. The users include educators and learners as well. Through a PA, an user can delegate rights to his/her agent, manage her/his profile, select his/her preferences and configure the options provided by the task agents. Users can meet their PAs simply by opening a secure web page in a desktop or laptop or pocket PC. For example, an Instructor PA is an agent of an instructor and acts as an assistant to the instructor, helping the instructor to design, generate, deliver, maintain e-learning courses and so on. To fulfil the tasks delegated by the instructor, the PA interacts with some tasks agents or other instructor PAs. Another example of PA is learner PA. A learner PA is an agent designed for a learner and acts as a companion or an assistant of the learner, helping the learner perform activities such as searching for learning objects (including courses), plan his/her study and so forth. To fulfil the tasks delegated by the learner, the learner PA interacts with some task agents or other learners' PAs in the environment.

PAs are deployed through regular website. A user can apply for an account from a web page and then login to and configure his/her PA through the GUI.

3.2 Task agents

A Task Agent (TA) is designed for performing certain specific tasks such as decision making or coordinating with other TAs or PAs. Because a TA is deemed a 'common resource' shared by many users and agents, its processing capability comprises a spooling function, in which requests are queued in accordance to their priority.

A TA runs on an agent platform. The agent platforms facilitate the creation, deletion and locating of the agents. They also involve inter-agent communication. One or more agent platform forms a distributed agent development and deployment environment. The agent management service of an agent platform registers the agents and assigns a unique agent identification to them and records agent information such as agent types.

3.3 Web services

The WS for an adaptive e-learning typically includes 'Knowledge Management Web Services' and 'Information Management WS'. Knowledge Management Web Services are designed to manage, locate, analyse and retrieve the knowledge resources. For example, to manage domain knowledge, a domain ontology web service is designed for providing services about a taxonomy database. Information management WS deal with 'learner information management', 'staff information management', 'course information management', 'learning objects management' and so on.

Through deploying Domain Model (Ontology) Web Services and Ontology mapping mechanisms, reuse and interoperability can be realised because the domain models can be shared between applications in the Semantic Web setting.

3.4 Interactions among the agents and WS

A PA interacts with a TA through sending the latter a request about a task to be done and expects to receive a reply from the TA.

A TA plays either the role of a client of a web service or the role of a supporter of a web service. As a client of WS, an agent can perform searches of different entries stored in a UDDI and can contain reason about the semantics of WS and mediate and compose WS.

A TA on an agent platform performs a 'find' operation on a service registry. The agent finds the entry for the service and uses the listed URL to download a copy of the WSDL. Using the WSDL, the agent generates a programme to serve as the Service Requester to access the service. When this is complete, the agent tests it by requesting that it performs a 'bind' operation on the web service. After the 'bind' operation is successful, the agent makes message- and RPC-style calls to a web service and waits for a response.

As a supporter of a WS, a TA facilitates and enables the web service. The WS benefits from the ability of the agents to perform the task *autonomously* and *intelligently, dynamic creation* of agents and semantic level communication. Most WS in adaptive e-learning environments can profit from the *flexibility, robustness, autonomy* and *intelligence* of agents. For example, we need 'spiders-like' monitoring agents to support course information WS by monitoring and maintaining web course materials (Lin and Poon, 2004). Another example is that vast educational resources stored in learning object

repositories available today and tomorrow simply will not be able to function without being able to delegate to agents the multitude of tasks that would otherwise be left to armies of people to handle (Lin et al., 2004).

4 An example: eAdvisor

4.1 Background

The MSc-IS program of Athabasca University are designed to maximise student flexibility and allow student control of place, pace and sequence of learning. All courses are delivered online with no residence requirements. This programme consists of 13 required courses allowing students to select from 16 optional career tracks. Maximising openness and choice can lead to learners making poor choices, thus, flexible systems need to be accompanied by strong advice, monitoring and support systems (Tait and Mills, 2003). The continuing challenge is to maximise the quality and availability of this support while continually reducing its cost. Given a wide number of courses choices or paths through the curriculum, it is important that each student be assigned an academic advisor who is a faculty member of the university. To be effective, advisors and faculty need to understand each individual student's educational and career objectives, their time constraints and their progress through the programme. These can then be matched against degree requirements, course start dates, course availability and prerequisites to generate a personalised course path. The provision of effective programme advice for the average of 30 graduate students per faculty member is a time-consuming task. To meet the students' needs and reduce the workload of the advisors, we have been developing an intelligent system – eAdvisor. The system is able to provide the students with an automated programme planning and scheduling service. On behalf of the students' advisors, the system will assist them in generating *up-to-date, personalised* and *optimal* study plans by constantly monitoring and utilising related knowledge and information resources.

Most research in adaptive educational hypermedia has focused on techniques for adaptation at navigational level and at content level (Brusiovsky and Nejdl, 2004). For example, Dolog et al. (2004) proposed *Smart Space for Learning,* which supports access to individualised learning materials and experiences using Personal Learning Assistants. At the programme planning level, the California State University has set the implementation of the PeopleSoft degree audit system at all campuses (Post, 2004) to actively monitor a student's course decisions during registration and to provide information that informs course decisions.

4.2 The architecture of eAdvisor

The components and their relationships are illustrated in Figure 2.

It consists of three kinds of agents, four WS, an agent platform and two relational databases. To take full advantage of WS and agent architecture we did not implement everything into one single application and did not run all of the functions on the same platform. The scheduling agent itself does not maintain/update any of the information it requires for the advising functionality. Instead, it will try to retrieve the information it requires. The scheduling agent keeps a list of end points of the WS that holds the

information. Namely, we have a web service for student information, including the background of the student, a web service for Course Repository and a web service for knowledge area modelled by using Protégé 2000.

- A PA in eAdvisor is a web application with a user profile database. The interface of the PA shown in Figure 3 allows a graduate learner to input her/his preferences (e.g. assessment styles) and parameters regarding programme planning and course selection (e.g. graduation date and credits to be obtained each semester) and manage the profile of the learner.
- The monitoring agent is responsible for monitoring the two databases. Whenever there are significant changes in the course schedules or learner information update in the databases the agent will notify the scheduling agent of the changes.
- The scheduling agent generates the individualised study plans in two ways. One is generating a plan according to the instant request from the PA of a learner. Another is that if the scheduling agent receives a notification message about the course schedule changes or learners' learning progress update from the monitoring agent, it will generate a plan for all related learners. Then, the scheduling agent will send requests to the notification web service to notify the learners and their advisor in the way specified by the learner, for example, through-mail.
- The Learner Information web service, supported by a Learner Profile database, includes requesting/providing contact information, performance, job objectives, background, programme status and so on. The Learner Profile database can be updated by the learners and administrators. We adopted the IMS Learner Information Specification (IMS LIP, 2005). There are 11 categories in the specifications. For simplicity and usability, we used four categories in the learner model. They are 'identification', 'goal', 'accessibility' and 'transcript'. We believe that these four categories are enough in this context of programme planning and scheduling.
- The Course Schedule web service is based on the Course Information database, providing information about course availability.

Figure 2 Architecture of eAdvisor

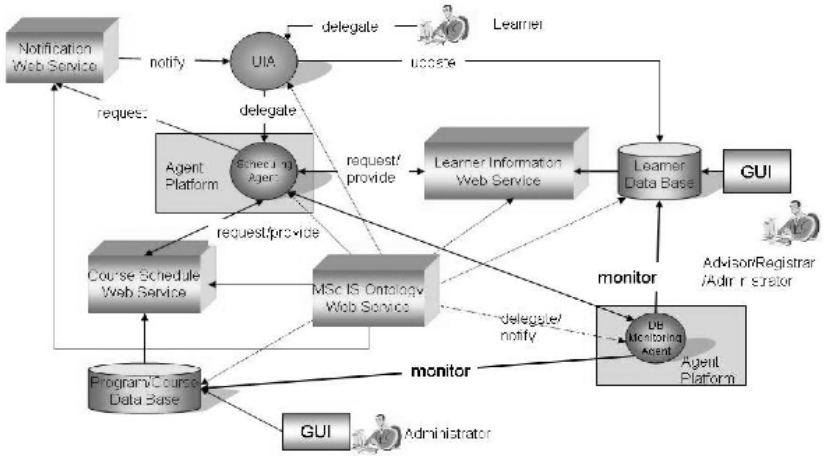
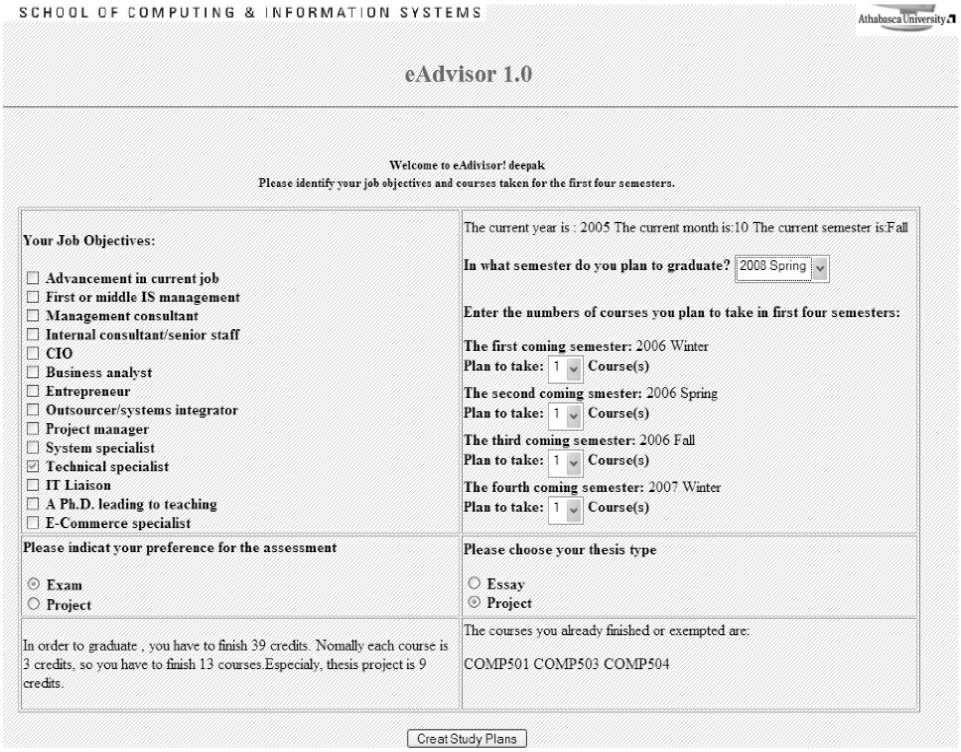


Figure 3 The interface of eAdvisor



All agents and WS mentioned above should explain the meaning of received messages in the same way. This is accomplished by sharing an ontology deployed as MSc IS ontology web service.

We use JADE (Java Agent DEvelopment Framework) (http://jade.tilab.com/) as the agent platform, which is a software framework to develop agent-based applications in compliance with the FIPA (the Foundations for Intelligent Physical Agents) (http://www.fipa.org/) specifications for interoperable intelligent multi-agent systems.

4.3 MSc IS domain ontology

MSc IS Ontology is based on MSIS 2000 (Gorgone et al., 2000) and describes the terms and their relationships, information about the programme structure and business rules. The MSc IS Ontology is modelled with Protégé 2000 (Knublauch, 2003) and represented by OWL (W3C, 2004). We developed an ontology web service. We created the domain model and data entry forms and used the instance tab to acquire instances of the classes defined in the ontology. Then, we populated the model with information so that the Protégé library can be accessed using a Java API to retrieve this information for use in the Java WS.

Online courses, being special kinds of learning objects, deal with one or more domain concepts in the MSc IS Ontology. The concepts are modelled based on the ACM Computing Classification System (Gorgone et al., 2000). The courses are indexed with

the metadata supplied by academics and course designers, following the IEEE LOM standard (IEEE LOM, 1999). Figure 4 shows the Course classes in the MSc IS ontology. Figure 5 shows a screenshot using Protégé 2000 to build the ontology.

Figure 4 Course classes in the MSc IS ontology

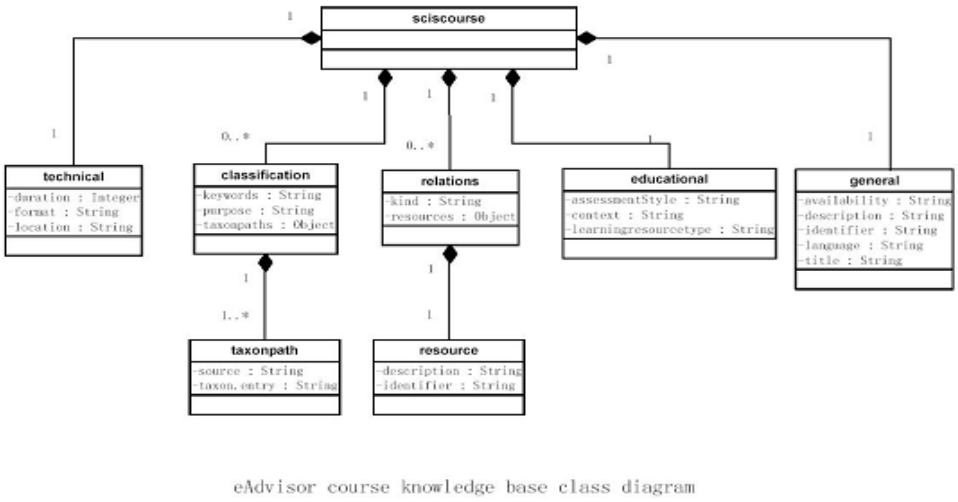
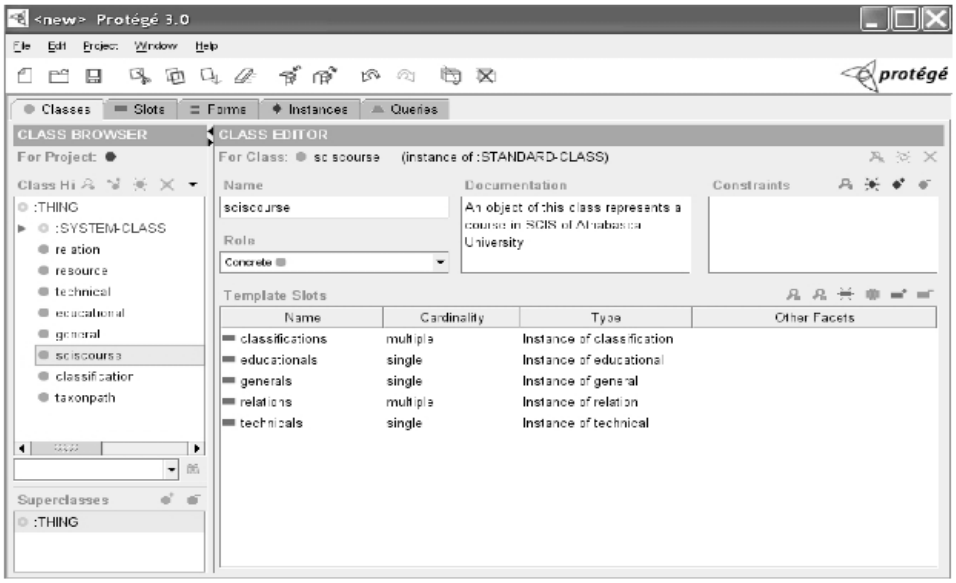


Figure 5 Screen shot using Protégé 2000



4.4 MSc IS course information web service

In eAdvisor, we built a web service to provide the course information. This web service is implemented based on the Apache axis and can be accessed via SOAP. We choose this architecture to greatly improve the flexibility of our system. Firstly, the course

web service can be managed/maintained by a different department (e.g. register department) rather than the department using eAdvisor. This makes it possible that several departments share their resources in a university. Secondly, the course web service can be accessed outside the Athabasca University, which means that similar advising systems in other universities can share the course ontology information from Athabasca University and then they can put Athabasca University's courses in their student's study plan.

There are three main operations in the web service: getCourseArray, getCourseTitles and getACourse. GetCourseArray is an operation, which can return all the course objects, each of which includes all the ontology information for a course at Athabasca University.

getCourseTitle is an operation, which just returns the title information of all the course objects, while getACourse returns a course object according to the inputted course title. The following is a part of the WSDL file of the course ontology service.

```
<wsdl:portType name="courseService">
        <wsdl:operation name="getCourseArray" parameterOrder="in0 in1">
<wsdl:input message="impl:getCourseArrayRequest" name="getCourseArrayRequest"/>
<wsdl:output message="impl:getCourseArrayResponse" name="getCourseArrayResponse"/>
</wsdl:operation>

        <wsdl:operation name="getCourseTitles" parameterOrder="in0 in1">
<wsdl:input message="impl:getCourseTitlesRequest" name="getCourseTitlesRequest"/>
<wsdl:output message="impl:getCourseTitlesResponse" name="getCourseTitlesResponse"/>
</wsdl:operation>

        <wsdl:operation name="getACourse" parameterOrder="in0">
<wsdl:input message="impl:getACourseRequest" name="getACourseRequest"/>
<wsdl:output message="impl:getACourseResponse" name="getACourseResponse"/>
</wsdl:operation>
</wsdl:portType>
```

Each course object returned from the course service is a complex java object. That java object must be serialised first and then it could be transferred to the client side over SOAP. From the following WSDD, you can find how a Java object is configured to be serialised in web service.

```
    <service
            name="courseService" provider="java:RPC">
    <parameter name="className"
            value="courseService.courseService"/>
    <parameter name="allowedMethods" value="getCourseArray,getCourseTitles,getACourse"/>
    <parameter name="scope" value="Application"/>
    <beanMapping qname="myNS:Course" xmlns:myNS="urn:courseService"
languageSpecificType="java:courseService.Course"/>
    <beanMapping qname="myNS:technical" xmlns:myNS="urn:courseService"
languageSpecificType="java:courseService.technical"/>
    <beanMapping qname="myNS:educational" xmlns:myNS="urn:courseService"
```

```
languageSpecificType="java:courseService.educational"/>
    <beanMapping qname="myNS:relation" xmlns:myNS="urn:courseService"
languageSpecificType="java:courseService.relation"/>
    <beanMapping qname="myNS:resource" xmlns:myNS="urn:courseService"
languageSpecificType="java:courseService.resource"/>
    <beanMapping qname="myNS:classification" xmlns:myNS="urn:courseService"
languageSpecificType="java:courseService.classification"/>
    <beanMapping qname="myNS:general" xmlns:myNS="urn:courseService"
languageSpecificType="java:courseService.general"/>
    <beanMapping qname="myNS:taxonpath" xmlns:myNS="urn:courseService"
languageSpecificType="java:courseService.taxonpath"/>
    <beanMapping qname="myNS:courseArray" xmlns:myNS="urn:courseService"
languageSpecificType="java:courseService.courseArray"/>
    </service>
```

The course ontology information retrieved from the course ontology web service is mainly used by the plan generating algorithm. The plan generating algorithm uses the ontology information to generate optimal study plans.

4.5 Notification web service

The notification WS makes use of the JavaMail class to perform the actual sending. The web service receives the sender or recipient information from the scheduling agent through sending XML request messages.

4.6 IMS LIP based student model and student information web service

eAdvisor traces the learner's profile by building a *Student Model*. We use the *IMS LIP 1.0.1* standard (http://www.imsglobal.org/profiles/index.html) suitably extended to represent the student's profile. IMS LIP 1.0.1 addresses the interoperability of internet-based learner information systems and supports the exchange of learner information. This model represents, in addition to the student's basic information, the student's skills concerning every MSc IS Ontology's curriculum building block and courses together with the student's job objectives and backgrounds. The representation of the student's profile is updated by the administrator every time the student completes an online course.

4.7 Programme planning and scheduling agent

The task of the agent is to provide a list of reasonable alternate programme plans for a student. Although the final decision will be up to the student, his academic advisor and the programme coordinator, the agent can respond to the latest changes in the environment and promptly advise the learner and his/her advisor. To facilitate the planning process, the agent can use the programme model, students' background and preferences and a list of extensible searching heuristics. The algorithm of planning and scheduling is out of the scope of this paper.

To achieve individualised course plans, the agent considers the courses the students have taken before. However, the presumption of this approach is that the courses can be mapped to the Knowledge Area in the MSc IS Ontology.

A *programme constraint* is a factor that affects a study schedule from the point of view of the programme. In our implementation, we have two constraints:

1 *Starting point*: a student must start with a certain course

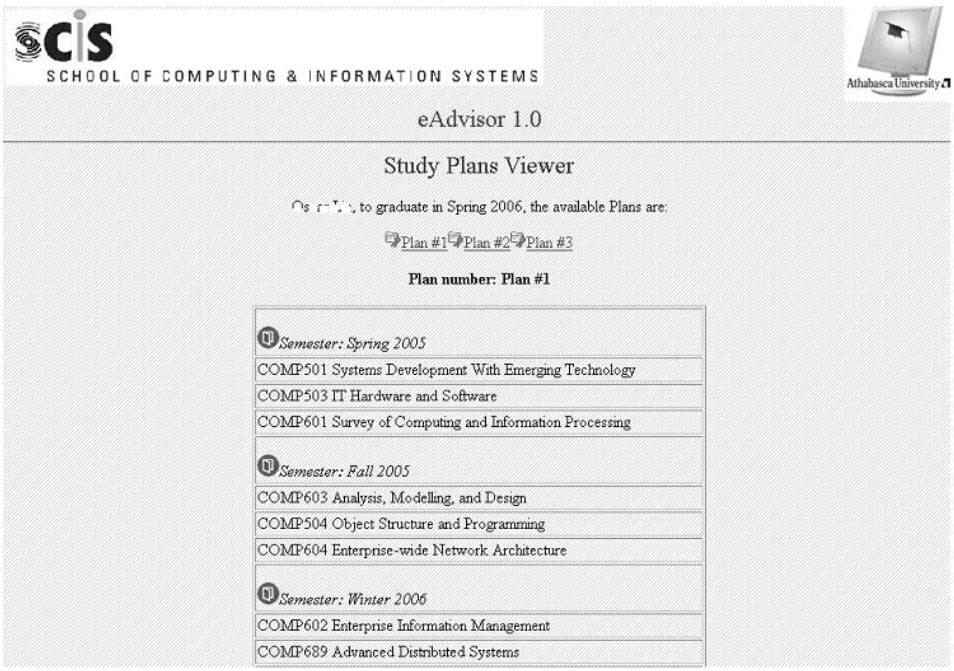
2 *Semester, minimum and maximum courses*: courses are grouped into a semester.

Courses taken in the same semester are considered as taking the courses at the same time. A student can take a minimum number of courses up to a maximum number within a semester. In our example, we prescribed that: $1 \leq$ no. of courses/per semester ≤ 2.

Programme constraints are not limited to the above two examples. Other conditions, such as course availability, the number of credits per course and financial constraints, also affect the schedule.

Heuristics for ranking the plans are introduced to achieve more reasonable solutions. Figure 6 shows an example of schedules generated by the scheduling agent in eAdvisor.

Figure 6 Schedule generated by the scheduling agent



5 Conclusions and future work

E-learning information standardisation, e-learning resource development, web technologies and AI technologies have paved a way for adaptive e-learning. To integrate intelligent software agents into existing legacy learning environments or heterogeneous

e-learning environments, one may encounter many difficulties. WS technology provides a new way to integrate existing systems or applications, and the ability to access data in a heterogeneous environment and to provide interoperability of components and learning content.

We have proposed an architecture for designing and developing adaptive e-learning environments by integrating agents and WS. Some agents and WS in the architecture have been developed for both real applications and experiments.

The Semantic Web is not only providing access to the contents of the web but also makes it possible to access WS that present certain tasks that can be done automatically. We will explore how to make agents able to find, composite, execute and monitor those WS under the control of the Semantic Web. OWL-S (OWL-based web service ontology) will be used, which supplies WS providers with a core set of mark-up language constructs for describing the properties and capabilities of their WS in an unambiguous, computer-interpretable form (OWL-S, 2004).

We have developed a prototype of an intelligent programme planning and scheduling system, eAdvisor, that will be able to generate timely, personalised and adaptive learning plans and schedules best suited for each learner in MSc IS of Athabasca University.

Our research is still in its initial stage. There is a lot work to do to make eAdvisor work well enough for practical use. For example, the schedule optimisation algorithm of the scheduling agent should be further improved. Further work also includes implementing and evaluating a functional version of eAdvisor with the agents described in this paper. More functionality could then be implemented and we should maintain a yellow pages service for the WS in the UDDI format so that we can eliminate the need of storing WS end points in the agents.

Acknowledgements

We would like to thank National Science and Engineering Research Council (NSERC) of Canada and Athabasca University for sponsoring this research project.

References

Aroyo, L. and Dicheva, D. (2004) 'The new challenges for e-learning: the educational Semantic Web', *Educational Technology and Society*, Vol. 7, No. 4, pp.59–69.

Baylor, A. (1999) 'Intelligent agents as cognitive tools for education', *Educational Technology*, Vol. 39, No. 2, pp.36–41.

Blackmon, W. and Rehak, D. (2003) 'Customized learning: a web services approach', *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003*, No. 1, pp.6–9.

Brusiovsky, P. and Nejdl, W. (2004) 'Adaptive hypermedia and adaptive web', in M. Singh (Ed). In *Practical Handbook of Internet Computing*, CRC Press.

Brusilovsky, P. and Nijhawan, H. (2002) 'A framework for adaptive e-learning based on distributed re-usable learning activities', in M. Driscoll and T.C. Reeves (Eds). *Proceedings of World Conference on E-Learning, (E-Learn 2002)*, Montreal, Canada, 15–19 October, AACE, pp.154–161.

Chan, T-W. (1995) 'Artificial agents in distance learning', *International Journal of Educational Telecommunications*, Vol. 1. Nos. 2/3, pp.263–282.

Chen, W. and Mizoguchi, R. (1999) 'Communication content ontology for learner model agent in multiagent architecture', *Proceedings of AIED'99 Workshop on Ontologies for Intelligent Educational Systems*, Available at: http://aied.inf.ed.ac.uk/members99/resources/ontologies/main.html.

Dicheva, D. and Aroyo, L. (2004) 'General architecture supporting component-based EIS interoperability', *Proceedings of the International Workshop on Application of Semantic Web Technologies for E-Learning (SW-EL'04), in conjunction with the Intelligent Tutoring Systems Conference (ITS'04)*, Maceiò, Brazil, 30 August–3 September, pp.21–28.

Dolog, B., Miklós, P., Olmedilla, Z.D. and Sintek, M.T. (2004) 'Conceptualizing smart spaces for learning', in T. Anderson and D. Whitelock (Eds). *Journal of Interactive Media in Education, Special Issue on the Educational Semantic Web*, No. 9, Available at: http://www-jime.open.ac.uk/.

Gorgone, J.T., Gray, P. and Feinstein, D.L. (2000) 'MSIS 2000, model curriculum and guidelines for graduate degree programs in information systems', *Data Base*, Vol. 31, No. 1.

Greer, J., McCalla, G., Vassileva, J., Deters, R., Bull, S. and Kettel, L. (2001) 'Lessons learned in deploying a multiagent learning support system: the i-help experience', *Proceedings of AIED'2001*, San Antonio, pp.410–421.

Hogeboom, M. and Lin, F. (2004) 'Developing domain model web services for agent-supported distributed learning using PROTÉGÉ 2000', *IEEE Learning Technology Newsletter*, April, Vol. 6, No. 2.

Huhns, M.N. (2002) 'Agents as web services', *IEEE Internet Computing*, July/August, pp.93–95.

IEEE LOM (1999) Available at: http://www.ischool.washington.edu/sasutton/IEEE1484.html.

IMS LIP (2005) Available at: http://www.imsglobal.org/profiles/index.html.

Kay, J. (1999) 'Ontologies for reusable and scrutable student model, position paper', *Proceedings of AIED99 Workshop on Ontologies for Intelligent Educational Systems*, Available at: http://aied.inf.ed.ac.uk/members99/resources/ontologies/main.html.

Knublauch, H. (2003) 'An AI tool for the real world: knowledge modeling with Protégé', *JavaWorld*, 20 June.

Mizoguchi, R. and Bourdeau, J. (2000) 'Using ontological engineering to overcome AI-ED problems', *International Journal of Artificial Intelligence in Education*, Vol. 11, No. 2, pp.107–121.

Moreale, E. and Vargas-Vera, M. (2004) 'Semantic services in e-learning: an argumentation case study', *Educational Technology and Society*, Vol. 7, No. 4, pp.112–128.

Lin, F. and Poon, L. (2004) 'Integrating web services and agent technology for e-learning course content maintenance', in B. Orchard, C. Yang and M. Ali (Eds). *Innovations in Applied Artificial Intelligence*, LNAI 3029, Springer, pp.848–856.

Lin, F., Esmahi, L. and Poon, L. (2004) 'Integrating agent technology and web services into distributed learning environments', in F. Lin (Ed). *Designing Distributed Learning Environments with Intelligent Software Agents*, IGP, pp.184–217.

Mitrovic, A. and Devedzic, V. (2004) 'A model of multitutor ontology-based learning environments', *International Journal Continuing Engineering Education and Lifelong Learning*, Vol. 14, No. 3, pp.229–245.

OWL-S (2004) Available at: http://www.daml.org/services/owl-s/.

Post, W. (2004) 'Building minds, not widgets: technology for the business of learning', *IEEE IT Professional*, September October, pp.12–18.

Razmerita, L., Angehrn, A. and Maedche, A. (2003) 'Ontology based user modeling for Knowledge Management systems', *Proceedings of the User Modeling Conference*, USA, pp.213–217.

Tait, A. and Mills, R. (2003) *Rethinking Learner Support in Distance Education: Change and Continuity in an International Context*, London: Routledge.

Thaiupathump, C., Bourne, J. and Olin Campbell, J. (1999) 'Intelligent agents for online learning', *JALN*, November, Vol. 3, No. 2.

W3C Ontology Web Language (2004) Available at: http://www.w3.org/2004/OWL/.

W3C Web Services Architecture Working Group, Working Draft 14 November 2002, Available at: http://www.w3.org/TR/ws-gloss/.