A Software Agent Based Searching Approach for Constructivist Learning Over the Internet

Weidong Pan, **Mao Lin Huang** and **Igor Hawryszkiewycz**
Faculty of Information Technology
University of Technology, Sydney
PO Box 123 Broadway, NSW 2007
Australia

Abstract. Finding out appropriate learning resources on the Internet is an important step in learning over Internet by using a constructivist method. Because the information available on the Internet grows rapidly, it is often difficult for a learner to search for a particular learning resource through navigating the large information sea. The use of commercial search engines can make the search much easier, but it is still difficult for the ordinary learners. This paper proposes the use of software agents to assist learners to find out required learning resources over the Internet. We present three approaches that the agent for an individual learner may use: 1) improve and optimize the search conducted by a commercial search engine, 2) seek advices from the teacher agents for appropriate learning resources and 3) seek suggestions for appropriate learning resources from the fellow student agents. Based on the analysis of the advantages and disadvantages of these approaches, a combined implementation is further proposed. Finally we describe the preliminary results and the outlooks of the research.

Keywords: learning resources, software agents, Internet, search, optimization, consulting

(Received March 29, 2005 / Accepted July 10, 2005)

1. Introduction

According to the constructivist theory for learning, learning is a process of constructing knowledge by a learner [8]. In such a process, an important step is to find out appropriate learning resources that can be used by individual learners to construct knowledge for solving the problem at hand. Because the information available on the Internet grows rapidly day and day, it is often difficult for a learner to search a particular learning resource through the navigation in such a dynamic information sea.

The investigation on assisting users to efficiently search information resources on the Internet for a particular requirement has a considerable long history, almost along with the inception of the WWW technology. A lot of achievements have been obtained in this area. The search engines, e.g. *Google*, *AltaVista*, *Yahoo*, are probably the most typical examples. They can help users to find out relevant information resources on the Internet based on the predefined keywords through their specific interfaces. However such commercial search engines cannot offer

learners much help in searching for a particular learning resource. The major reasons can be summarized into threefold: 1) these search engines have been designed for general-purpose search of information resources, rather than for providing specialized services of searching learning resources; 2) they can only search public Web sites on the Internet, which means that they are unable search and find out learning resources residing at internal databases; 3) they restrict users only to enter key words to express their requirement for the information resources, which is not sufficient for learners to express their interests for searching required learning resources.

This paper proposes the use of software agents to assist individual learners to find out required learning resources. We suggest three approaches that the agent assigned to an individual learner may use: 1) performing the necessary improvement and optimization to the search conducted by a commercial search engine, 2) seeking advices for the learning resources from the teacher agents and 3) seeking suggestions for the learning resources from the fellow student agents. The paper is organized as follows. In the next section the framework of the multi-agent architecture for assisting individual learners to develop new competences by using constructivist learning methods is briefly described and the motivations for the current research are outlined. In the three sections followed, the three approaches are presented respectively, along with the relevant implementation techniques. In section 6, the advantages and disadvantages of the three approaches are analysed, and a combined implementation is proposed. In the last section, we conclude with a description of the on going and the future work on the research.

2. Using software agent technologies to assist learners to construct knowledge

Constructivists believe knowledge is individually constructed and socially co-constructed by learners through interaction with their environments [6]. A rich learning environment is considered as a major goal in constructivism where learners are engaged in active manipulative, constructive, intentional, complex, authentic, cooperative and reflective learning activities [4]. Constructivist environments supply learners with opportunities to construct new knowledge based on prior one from authentic experience. Learners are encouraged to confront problems full of meanings. In solving these problems, learners are facilitated to explore possibilities, invent alternative solutions, collaborate with others, try out ideas and hypotheses, revise their thinking, and finally present the best solution they can derive [4].

In order to assist learners to build new knowledge by using a constructivist method, we have concentrated our research on incorporating software agents into the learning environments to supply services for learners to facilitate knowledge construction. A multi-agent architecture has been configured to implement the services. The architecture consists of a number of agents with various expertise and they can be classified into three categories; i.e. learner agents, teacher agents, and tool agents. A learner agent is an agent that works for an individual learner to assist him in study. A teacher agent is one that works for a teacher to assist him in supplying various support services to the students. The third type of agents, called tool agents, are specialized agents that are used for supplying support services to the learning activities involved in the procedure of knowledge construction. The typical tool agents include email agent, navigation agent, BBS agent, Chat agent, etc. An important one among the tool agents is the facilitator agent, which must be emphasized here. The facilitator agent is the one that manages all the agents on the environments. It stores the information of all the agents in its database, with a record for an agent. The information includes their symbolic names and resident addresses and as well as the services they have registered to provide. All the agents in the environments work together to coordinately assist individual learners to develop new skills by using a constructivist method.

An important work in the learning on the Internet by using a constructivist method is to find out appropriate learning resources on the Internet for solving the problem at hand. An immediate way to assist individual learners in the work is to use a commercial search engine to perform the search for them. Because of the three reasons we summarized previously in this paper, the search result will contain a lot of unsuitable resources. There are two strategies to tackle the problem; one is to develop a new search engine specialized for search learning resources on the Internet and the other is to rely on a search engine to crawl and search on the Internet and in the meantime to employ an agent to improve and optimize the search process and result. We have chosen the latter one because of the success of the commercial search engines in searching information resources on the Internet [9] and the limitations of our available resources.

Moreover, we have discovered, from the investigation on the methods and processes of constructivist learning, that learners may seek for appropriate learning resources not only through their own search on the Internet but also from other channels. For instance, they may ask for advices on the learning resources from the teachers who teach the subject covering their learning requirements or ask for suggestions for the learning resources from their fellow students who have experienced the relevant study. This has aroused us to develop other ways to assist individual learners to find out required learning resources. Since every teacher who provides teaching services over the Internet can be assigned an agent for assisting his teaching services, the agent can on his behalf take actions [9], including automatically responding to the requests to the teacher from his students. It thus is possible to attain assistance from a teacher by consulting his agent. Based on the similar reason it is possible to get information from a student by communicating with the agent for him. Accordingly, two more approaches to assisting individual learners to find out required learning resources have been developed; one is to seek advices on the learning resources from the teacher agents and the other is to seek suggestions for the learning resources from the fellow student agents.

In the following sections these three agent-based approaches will be presented respectively.

3. Optimizing the search process and result of a commercial search engine

An agent, assigned to an individual learner, is expected to be responsible for assisting the learner to find out appropriate learning resources on the Internet according to his unique learning requirements and learning characteristics. We have developed three approaches for the agent to find out required learning resources. As the first approach, the agent dispatches a commercial search engine, e.g. G*oogle*, to conduct the practical search, and in the meantime it performs an improvement and optimization to the search process and result. By using this approach, the agent first translates the learning requirement into the keywords appropriate for the search and then adds some extra information to direct the search process. After the search criterion is determined, the agent forwards it to the search engine to search Web pages based on it. While the search result is returned, the agent collates the result. It first eliminates the Web pages from the result that are recognized unsuitable for learning through the clues, such as title, summary, URL, etc. It then downloads and parses those left pages with the ranking of relevance for further eliminating of Web pages whose context does not match the learning characteristics of the learner. Finally the agent has attained a list of the Web pages that are most appropriate for its owner learner. The working process of this approach is depicted in Figure 1.

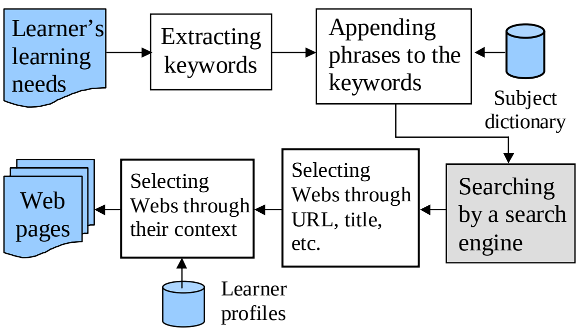

Figure 1. The working process of method 1

As described above, the agent improves the search process and optimizes the search result through the pre-process and post-process. This strategy is similar to those of most previous researches on the improvement of searching information resources over the Internet, e.g. Inquirus [2], SavvySearch [3], and MetaCrawler [7], and the automated question answering system based on the search on the Internet, e.g. AnswerBus [10], etc. However, the method for the pre-process and post-process used here is distinct from others in that we use *subject dictionary* to provide extra information while determining the search criterion. In addition, we parse the full text of the Web pages to match the learning characteristics of individual learners.

3.1 Search criterion determination

The search criterion is determined by converting the key words into a set of search criteria to narrow the focus of search into a small set of the information resources, so that the search result returned from the search engine could be more relevant to the learning requirement. It is implemented through two classes of modifications conducted by the agent: 1) utilization of search engine-specific options and 2) appending extra information to the keywords. The agent performs a lookup table in the *subject dictionary* and captures a few phrases related to the keywords from it. It then appends them to the keywords and forwards their combination to the search engine, as well as the suitable settings for the search engine specific options, e.g. phrase search, AND/OR choice, domain, language, etc.

It can be seen from Figure 1 that the *subject dictionary* is a crucial component in the determination of search criterion. The *subject dictionary* is designed specifically for expressing a hierarchical architecture of the contents in a subject. It is similar to the index of a book. In the index of a book, under a chapter there is a list of names of the sections that belong to that chapter. However, in the *subject dictionary* for a subject, within a chapter there is a list of phrases that should appear in the context of that chapter. The main idea here is to use those phrases to constrain the chapter names while they are used as the keywords in search. The *subject dictionary* for a particular subject can be developed by refining the index of a book for the subject. The refinement is, for every part of the subject, to specify the relevant phrases that should appear in its context through the investigation on the hierarchical architecture of the subject content.

Let's use a simple example to illustrate this method. As it is well known that the update of a database involves insertion, deletion and/or modification of data records.

Therefore, the phrases under "database record update" in the *subject dictionary* for subject *database* should contain "insert", "delete" and "modify". Now let's assume that a learner wants to learn how to update database and the keywords extracted from the learning requirement is "database record update", the agent then obtains the phrases "insert", "delete" and "modify" through a lookup table in the *subject dictionary*. It then uses "database record update" AND "insert" AND "delete" AND "modify" as the search criterion and contacts *Google* to perform the search. The experiment conducted on 12 March 2004 indicates that 9 results are returned from *Google* for the above search. In comparison, 822 results are returned if we only use the phrase "database record update" as the criterion in the search; and about 2,890,000 results are returned if we just enter "database record update" (without two quotes) into the search box without doing any settings.

As shown in the simple example, more valuable results can be obtained through appending the appropriate phrases to the keywords than only using the keywords as the search criterion. This appending of phrases, together with other considerations in determining the search criterion, enables the search engine to return more valuable resources related to the learning requirements.

3.2 Context match

The objectives for downloading and parsing a Web page are twofold: 1) to check its availability because the information resident at the Internet changes dynamically and some Web pages retrieved by a search engine are likely no longer available and 2) more importantly, to further examine whether the context of a Web page can suit particular learning characteristics of a learner. The motivation behind is that learners can be best engaged by the Web pages whose context can be matched with their particular learning characteristics.

To determine whether the context of a Web page can suit an individual learner, it requires a model to describe learning characteristics of individual learners. Many previous studies have been concentrated on the quest to build a model of learning characteristics. However, it is fair to say that it has not been quite clear which aspects of learning characteristics are worth modelling [5]. In general this should consider such factors as learning style, background, skill level etc. The agent for an individual learner has maintained the information about these factors in the learner's profile and this information is dynamically updated as the learning progresses. The challenge here is to determine the learning characteristics based on the information in the profile and then to take them as a criterion to judge if a Web page suits a learner.

For this purpose, the relevant learning characteristics of an individual learner are described through a set of learning demands in our current implementation. This set includes various desires for a learning resource. As it can be seen, an academic topic is likely to be presented through different contexts in Web pages, e.g. some may use text with graphics, tables, animations, or even video movies to present it, whereas others may only use text; some may present it in a detailed mode e.g. using examples, whereas others may use a concise mode to present it; some may present it along with related reasoning procedure e.g. quantitative analyses, deduction of formulations, proof of theorems, whereas others may present it without any extra explanations; some may attach an index, an abstract, a summary, a reference, or a few exercises or questions, whereas others may do not, etc. We believe that it will benefit the learning to match the presentation mode of the Web pages with the individual demands for learning resources. Thus, we take the demand set as the criteria to select Web pages.

While judging whether a Web page suits an individual learner, the agent of the learner analyses the full text of the Web page. It iteratively checks whether each of the learner's demands can be satisfied by the context of the Web page and sums the number of the satisfactions. A Web page is considered as suitable for the learner if the sum of the satisfactions is larger than a predefined threshold value.

As the first step to pursue the matching between the context of Web pages and the learning characteristics of individual learners, the current implementation is simple but very efficient. It is easy to further extend and optimize the set of learning demands that towards more accurately describe the learning characteristics of individual learners.

4. Seeking advices from the teacher agents

As the second approach, the agent that has been assigned to an individual learner assists its owner learner to find out required learning resources through consulting the agents for the teachers who are teaching the subject covering the learner's learning requirement. First, the agent extracts the keywords from the learning requirement. It then obtains a subject that covers the learning requirement from the *subject dictionary* based on the keywords, and then consults the facilitator agent for the information about relevant teachers who are teaching the subject. After having received the information from the facilitator agent, it initiates dialogues with the agents associated with teachers to seek advices for appropriate learning resources. These agents, based on the subject information, respectively retrieve the related learning resources from the knowledge base, and send back a list of the Web pages containing recommended learning resources. The learner's agent collects the information about the learning resources from each of teacher's agents respectively, and then orders them by the recommended rates. Finally, the learner's agent presents the learner a list of Web pages with highest recommended rates. The working process of this approach is depicted in Figure 2.

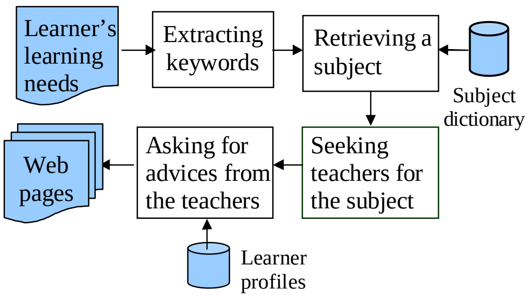

Figure 2. The working process of method 2

It can be seen from the above description that communications between agents are most important in this approach. Without efficient and flexible exchanging of messages among agents, the learner's agent will not be able to get enough information to generate recommendations of learning resources for the learner. The facilitator agent also plays an important role in the process.

4.1 Communications among agents

The communications among agents have been recognized as an active focus of research since the early days of software agent technology and a number of impressive achievements have been obtained. Among them a well-known one is the Knowledge Query and Manipulation Language (KQML) [1]. KQML defines both a message format and a message-handling protocol that support run-time information exchanging and knowledge sharing among agents [1]. Since KQML has been successfully applied into many agent-based systems with different architectures and it allows us to use any of the standard communication protocols e.g. TCP/IP, email, HTTP, etc. to implement the communication at the transport level, we have employed KQML as the communication language for the agents to exchange messages.

We have implemented a variant of KQML with a core set of KQML performatives for the information exchange. In the current implementation, we have made some slight modifications to the parameters of the KQML performatives for the simplicity and high efficiency. For instance, the field *language* has been dropped because all the agents use only a unique language, to exchange messages. Also the field *force* has been dropped to simplify the implementation. A new field, *dialog*, has been added to improve the efficiency of communication between agents. This is for such a kind of communications where two agents need to sequentially exchange messages one by one. By the *dialog* field, the path between the two agents is likely to be constructed only once and after it is constructed, all the messages between the two agents might be transmitted through it.

KQML performatives have defined the permissible communication actions for the agent to take. They are not aware of the message content that they carry and thus, there are no restrictions on the language to express the message content. In our implementation the message content is encoded using the Structural Query Language (SQL). SQL provides the syntax, semantics, and a set of useful operators for the content expression, yet it is not concerned with the concrete meanings of contents.

As an example, suppose that the agent for learner *i* wants to ask the facilitator agent for the information about the teachers teaching subject *XXX*. This can be implemented through the following two messages:

```
(ask-one
  :sender learner-agent-i
  :receiver  facilitator-agent
  :reply-with q-1
  :dialog d-1
  :content ( Select teacher where subject="XXX"))

(reply
  :sender  facilitator-agent
  :receiver  learner-agent-i
  :in-reply-to q-1
  :dialog d-1
  :content (=Select teacher where subject="XXX"))
```

The first message, sent from *learner-agent-i*, uses performative *ask-one* to ask the facilitator agent for the information about *the teachers who teach subject XXX*. While the facilitator agent receives the message, it parses the *content* field to identify the meaning of the message and thus understands the message is a request for the information about *the teachers who teach subject XXX*. The *Select-where* is a SQL statement. The facilitator agent finds the required information through retrieving the registry of agents in the database and consulting other related agents. It then composes the second message, using KQML performative *reply*, to respond the request from *learner-agent-i*. While receiving the message, *learner-agent-i* knows it is the response to its request pertaining to *the teachers who teach subject XXX* because the value of the filed *in-reply-to* is equal to the value of the filed *reply-with* in the message it has sent. It then parses the *content* field and acquires the required information.

4.2 The facilitator agent

The facilitator agent is crucial in the implementation of the required communications because it is the facilitator agent who helps a learner agent to attain the information about the teacher agents. Without the support of the facilitator agent, the agents could not successfully exchange messages.

In our implementation, the facilitator agent is announced to all the agents while they are established. Every agent knows it and registers in it while being launched. The facilitator agent stores the information of all agents in a database, includes their symbolic names and real IP addresses, as well as the services they have registered to provide. Thus, it knows every registered agent about their needs and the services they can provide for others.

The facilitator agent listens at the standard port for the incoming messages and provides communication services. These services include forwarding messages to named services, routing messages based on the interpretation of message content, providing "matchmaking" between information providers and consumers, and providing mediation and translation services [1].

5. Seeking suggestions from the fellow student agents

As the third approach, the agent that has been assigned to an individual learner assists the learner to find out required learning resources through consulting the agents for the fellow students who have experienced the relevant study. By using this approach, the agent first extracts keywords from the learning requirement. It then performs a lookup table in the *subject dictionary* based on the keywords and attains a subject that covers the learning requirement. It consults the facilitator agent for the information about the learners who have experienced with the relevant study. After having got the information, it starts to dialogue with the agents for the fellow students to seek suggestions about the appropriate learning resources. The agent sends its owner learner's learning characteristics, e.g. background, interest, style, motivation, capability, etc. to the agent associated with one of the fellow students. After receiving this information, the latter compares the learning characteristics with its owner's. If both can be matched, it will retrieve the relevant learning resources from the learning history records in its knowledge base, and then send back a hyperlink pointing to the suggested learning resource. In this way, the agent of the learner gathers the learning resource information from several fellow students, and then orders them by the recommended rates. Finally, the agent presents the learner with a list of Web pages with the top recommended rates as the learning resources.

The implementation of this approach is similar to the second approach.

6. A combined implementation of the three approaches

Each of the three approaches presented above has advantages and disadvantages respectively. The first approach can be possibly used to help learners find out the up-to-date learning resources on the Internet. However, this approach has difficulties to target those learning materials that exactly suit the needs of individual learners. The major difficulties are in determining whether a Web page is most appropriate for the purpose of learning, and whether the context of a Web page suits an individual learner. This is because both determinations require the fully understandings of the Web page context. It's same as the understanding of natural languages which is still an open problem for research. Our current implementation is parsing the semantics of the Web page, and thus the result is not quite satisfied. Additionally it is not possible to find the learning resources resident at internal databases by using this method because it is based on the search conducted by a commercial search engine.

By using the second approach, agents can help learners to find out required learning resources without performing any practical search on the Internet. This will significantly reduce the consumption of the Internet resources. Moreover, by using this approach, learners can find the learning resources not only on the public Web sites but also at the internal databases, because it is likely for a teacher to recommend a learning resource that resides at an internal database. The problem is, however, that learning resources found by this approach are rarely the up-to-date information, and sometimes no longer available on the Internet.

The third approach also has the advantages as the second one, i.e. it is possible to find required learning resources on both public Web sites and internal databases without performing any practical search on the Internet. Furthermore, using this approach can find more suitable learning resources for individual learners than using the second one. This is because the suggestions about the learning resources are from a learner's fellows who have the same learning characteristics as the learner. Comparing with the second one, however, it takes time and consumes communication resources to find the fellows with particular learning characteristics.

In order to take the advantages and avoid (or minimize) the problems of these approaches, a combined implementation of three approaches is probably better than any single one. In this combined implementation, the learner's agent first consults the agents for the teachers who teach the subject to seek advices for the learning resources. After having received a list of Web pages from the teacher agents, it then starts the dialogues with the fellow student agents to ask for their comments to the list. The agents for the students remedy the list according to their owner' learning history records, by eliminating the Web pages that they believe unsuitable for the learning, changing the order of the list, and adding new Web pages that they think suitable for the learning. After having received the feedback from each of the fellow student agents respectively, the agent for the learner ranks the Web pages into a list. It then further checks the Web pages in the list to eliminate the ones currently unavailable on the Internet. Finally it presents the learner with the remained Web pages as the learning resources.

7. Conclusions

Finding appropriate learning resources on the Internet is a crucial step to conduct learning on the Internet by using a constructivist way. Because the information on the Internet grows rapidly and will continue to grow, it is extremely valuable to use technological means to assist individual learners to find out required learning resources based on their unique learning requirements and learning characteristics. In this paper, software agents are used to undertake the work. The preliminary research and experiment have verified that the agent for an individual learner can employ any one of the three approaches, presented in this paper, to assist its owner learner to find out required learning resources. The combined implementation of the three approaches can further void some possible problems in using any single approach.

The research is a part of a larger project for using software agents to assist learners to develop new competences by using a constructivist method, which is currently on progress. We will be pursuing constructivist learning environments, where learners may work together and support one another as they use a variety of tools and information resources in their guided pursuit of learning goals and problem-solving activities [8]. Software agents will be seamlessly integrated into the environments to actively provide learners with a wide range of support services. This research is to assist learners to find out appropriate learning resources for building knowledge for solving the problems at hand. Like other components developed by using software agent technologies, the implementation of this research will be integrated into the environments as a middleware to assist learners to construct knowledge by using a constructivist way. Also we will extend the *subject dictionary* and optimize the expression of learning characteristics for individual learners.

8. References

[1] Finin, T., Fritzson, R., McKay, D. & McEntire, R. (1994) "KQML as an Agent Communication Language". *The 3rd International Conference on Information and Knowledge Management (CIKM' 94)*. Gaithersburg, Maryland, USA.

[2] Glover, E. J., Lawrence, S., Gordon, M. D., Birmingham, W. P. & Giles, C. L. (2001) "Web Search--Your Way". *Communications of the ACM, 44*(12): 97-102.

[3] Howe, A. E. & Dreilinger, D. (1997) "SavvySearch: A Meta-Search Engine that Learns which Search Engines to Query". *AI Magazine 18*(2): 19-25.

[4] Jonassen, D. H. & Rohrer-Murphy, L. (1999). Designing Constructivist Learning Environments. In C. M. Reigeluth (Ed.), Instructional Design Theories and Models: a New Paradigm of Instructional Theory. MahWah: Lawrence Erlbaum Associates, Publishers, 1999, Vol. II, 215-240.

[5] Kelly, D. & Tangney, B. (2002) "Incorporating Learning Characteristics into an Intelligent Tutor". *The Sixth International Conference on Intelligent Tutoring Systems (ITS2002),* Biarritz, France and San Sebastian, Spain, June 2-7.

[6] Mayer, R. E. (1999) “Designing Instruction for Constructivist Learning”. In C. M. Reigeluth (Ed.), Instructional Design Theories and Models: a New Paradigm of Instructional Theory. MahWah: Lawrence Erlbaum Associates, Publishers. pp. 141-159, 1999.

[7] Selberg, E. & Etzioni, O. (1997) "The MetaCrawler Architecture for Resource Aggregation on the Web". *IEEE Expert, 12*(1), pp. 8-14.

[8] Wilson, B. G. (1996) "*Constructivist Learning Environments: Case Studies in Instructional Design*". New Jersey: Educational Technology Publications.

[9] Wooldridge, M. (2002) "*An Introduction to Multiagent Systems*". England: John Wiley & Sons, Ltd.

[10] Zheng, Z. (2002) "AnswerBus Question Answering System". *Human Language Technology Conference (HLT 2002)*. San Diego, CA, USA, March 24-27.