Ontology-based Learning Content Recommendation

Li-ping Shen* and Rui-min Shen

Department of Computer Science, Shanghai Jiao Tong University,
No.1954, Hua Shan Road, Shanghai 200030, China
Fax: 0086-21-62933205 E-mail: lpshen@sjtu.edu.cn
Fax: 0086-21-62932947 E-mail: rmshen@sjtu.edu.cn
*Corresponding author

Abstract. A new era of e-learning is on the horizon, hundreds of Learning Contents are created and more and more people begin to acquire acknowledge thru e-learning. The traditional teaching method is already showing its limitations that students from different backgrounds are still given the same contents at the same time, and they may only interest in part of a whole learning content. In this paper, we propose a novel way to organize learning contents into small "atomic" units called Learning Objects so that they could be used and reused effectively. The Learning Objects together with their ontology are systemized into knowledge base. An intelligent recommendation mechanism based on sequencing rules is then introduced with detail, where the rules are formed from the knowledge base and competency gap analysis. Finally we establish a test knowledge base system, using and extending the ontology editor Protégé-2000 and its Protégé Axiom Language.

Keywords: Learning Object, Knowledge Base, Recommendation, Competency Gap Analysis, Sequencing Rule, Protégé-2000.

Reference to this paper should be made as follows: Shen, L-P. and Shen, R-M. (2005) 'Ontology-based Learning Content Recommendation', Int. J. Continuing Engineering Education and Lifelong Learning, Vol. 15, Nos. 3~6, pp.308–317.

Biographical notes: Li-ping Shen is a Lecture in the Department of Computer Science of Shanghai Jiao Tong University. She received her master's degree in Electrical Engineering (Circuits and System) from Shanghai Jiaotong University in 2000 and now is a PhD candidate in Computer Science and Technology. She is also a member of the content based working group of the Chinese e-Learning Technology Standardization Committee (CELTSC). Her research interests lie in the area of computer networking and intelligent e-learning system. She has published papers in several conference proceedings and journals.

Rui-min Shen is a Professor in the Department of Computer Science of Shanghai Jiao Tong University. He is also a member of the committee of the CELTSC. His research interests are in the areas of data mining, e-learning technologies and information processing. He has published papers in many major journals and conference proceedings.

1. Introduction

A new era of e-learning is on the horizon with a huge market, a market and technology that encompasses Learning, Training, Marketing, and online Support, and almost everything hitting us electronically can be called eLearning, people have accepted e-Leaning to a great extent. And the traditional teaching method is already showing its limitations that students from different backgrounds are still given the same learning contents at the same time, and they may only interest in part of a whole learning content. Most eLearning contents today are organized in a linear, sequential way, with large number of pages and without any description per se. Such contents couldn't be used to provide the exact content to the learners and also couldn't be reused by other authoring tools and content management systems. In this paper, we propose a novel way to organize learning contents into small chunks called Learning Objects (LO) ---the "atomic" units of knowledge. There are synonyms of LO as Knowledge Object (Li and Close, 2000), Sharable Content Object (SCO) (Dodds, 2004) and Learning Asset (Charalampos et al., 2001). Reusable objects enable us to put appropriate knowledge in the direct path of users. We need to create instructionally sound objects that are usable inside and outside of the traditional learning environment in order to have a significant impact on business performance. It must be flexible and easily managed so that the granules can be reused in different applications and for mass production.

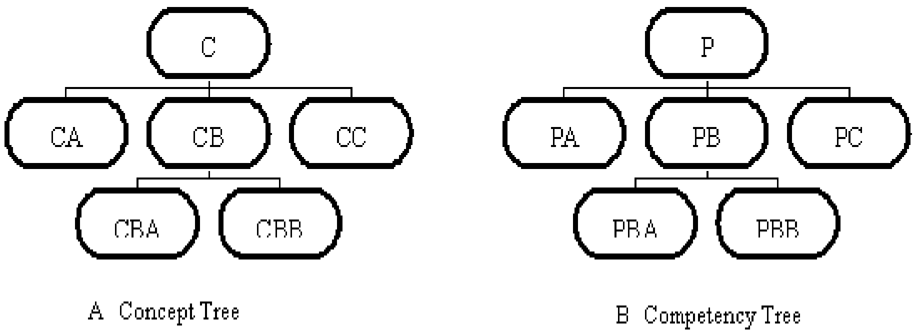
People have made great efforts on research about recommendation systems using content based filtering and collaborative filtering (Zeng et al., 2002). These systems locate and retrieve information with respect to users' profiles (interests and behaviors). Matching information with user profiles has been extensively studied in information retrieval and recommendation areas (Frakes and Baeza-Yates, 1992; Salton, 1989; Resnick and Varian, 1997). Most of the existing matching approaches is based on distance and similarity measures or probabilistic methods (Salton, 1989). Usually, the designer should select several features and design some similarity calculation or evaluation methods beforehand. With these features and methods, users can easily find useful information. However, because user profile(s) or similarity is predefined by the features and computing methods employed. Addition of new features or changes to existing user profiles can involve a significant amount of recomputing. This drawback means that the traditional matching approaches are not well suited to a dynamic environment where the users' preferences are likely to be subject to change. Another problem is that it is not easy to characterize and represent user profiles accurately, or to design correct methods to compute the similarity between information and user profiles precisely. In consequence the resulting recommendations may inevitably include some users with mistaken and uninteresting information.

This paper proposes to recommend learning content based on the expert learning object knowledge base and personal learning progress. The expert learning knowledge base incorporates information about simple sequencing. This recommend mechanism needs not the whole learner profile including interests and preferences, but just needs the information of the learning progress and learner's competency, so avoids the drawbacks explained above. In the following sections we will introduce the recommendation approach in detail. Section 2 introduces the method to establishing Learning Objects knowledge base by using the ontology editor Protégé-2000 (http://protégé.stanford.edu). The intelligent recommendation mechanism based on the Learning Object Knowledge Base and learner's competency is presented in section 3. We establish a test LO knowledge base by using and extending the ontology editor Protégé-2000 and its Protégé Axiom Language and it is described in section 4. Section 5 concludes this paper.

2. Learning Objects Knowledge Base

What is a Learning Object? How small should it be? During content design and authoring activities, when determining the size of a SCO, thought should be given to the smallest logical size of content that one might desire to have tracked by a LMS at run-time (Dodds, 2004). In order to use and reuse the LO efficiently, we define here a LO is a learning content lasting less than 30 minutes and containing only one knowledge point, i.e. concept.

Figure 1 Concept Tree and its corresponding Competency Tree



First of all, we need to define the contents that will be stored in the knowledge base (Karagiannidis, 2001): the concepts which are to be communicated, and their classification and inter-relation, i.e. the ontology of the content. This description can be performed by a domain expert, and does not require that learning issues are taken into account. For example, the concept "router" is the sub-concept of "networking device", and relates to "switch" and "gateway". All the concepts interrelated with each other build up a concept graph. In this paper, we only consider simple relationships that link concepts to a tree as Figure 1. We also need to define the dependencies between the concepts, like "prerequisites" i.e. defining the expertise required before presenting a specific concept to the learner. For example, "internet protocol" is the prerequisite of the concept "router". The competencies that are related to each concept in the ontology of the learning Objects are also defined. In this paper we suppose that each concept corresponds to one competency, so all the competencies form a competency tree as its corresponding concept tree as Figure 1. For each competency in the ontology, we define some questions and tests which determine whether the learner has understood the concept and should be granted a specific competency.

Subsequently, we need to compile and define the LOs that are instances of each concept of the ontology (only the leaves of the concept tree have LOs). LOs can be available as text files, images, videos, simulations, etc. Each LO is described through meta-data, such as how much it costs to see this information, the style and format it will be presented in, the language it uses, and the specific knowledge points in the ontology it is “linked” to. In order to describe the atomic units of knowledge concisely, we choose a subnet of the Chinese e-Learning Technology Standardization Learning Object Metadata Specification (http://www.celtsc.edu.cn/download/CELTS-3.1 (CD1.6).zip) and add some extensions such as the format element. The format element is used for Medium-Neutral Delivery, which allows us to select and modify the LO to any designated export format, based on what is most appropriate and cost effective. The sub-element resolution/track indicates the resolution of the figure and video LOs or the tracks of the sound LOs. The version element is used for rapid and automatic online update.

3. Recommendation Mechanism

In order to recommend appropriate content to different learner, we need to construct an application Knowledge Base on the LO Knowledge Base. Our recommendation Mechanism bases on competency gap analysis and the IMS simple sequencing specification (http://www.imsproject.org/simplesequencing), so we should further define the learner profiles and sequencing rules. We here just define simple learner profiles, for example, name, ID, learning progress, competency, objective competency and preferable language etc.

3.1 Sequencing Rules

The sequencing rules form the basis for the intelligent recommendation system, which define how the learners will navigate the concepts of the ontology, how different LOs are selected for different learners based on the learner profiles and the internal relationship of the concepts. According to the IMS simple sequencing specification (http://www.imsproject.org/simplesequencing), the sequencing rules have the following form: if constraints then action. The constraints may be one of (each could add unary operator NOT): satisfied, objective measure greater, completed, time limit exceeded etc., and the action may be: continue, retry, previous, skip and exit. We integrate the rules with competency gap analysis, i.e. before selecting the LO to be presented, system compares the competencies that the learner has acquired (referred as learner competencies following) and the objective competencies, computes the target competency, then thru the competency to concept binding and concept to LOs binding, present the selected target LO to the user. The objective competencies may have multiple competencies and is relatively stable during a learning period whilst the target competency is used to determine the next LO to be presented, so only has one competency and is always changing during the learning period. The following exhibits some examples of the sequencing rules. The initiate rules are used when the user logins to the system and the other rules are used during the learning process.
Initiate rule:

- If objective competencies not met then {compute target competency, present LO binding to target competency}.
- If objective competencies met then exit.

Sequencing rules:

- If satisfied then {update learner's competencies, compute target competency, present LO binding to the target competency}.
- If flat then represent the current LO.
- If failed then present the LO binding to one of the prerequisite.
- If objective competencies met then exit.

3.2 Target Competency Computation

The computation of the target competency in the sequencing rules is to find a competency that helps the user gain upon his objective competencies. The method is firstly to compute the gap competencies thru comparing the learner competencies and objective competencies, then to find a competency in or outside the gap competencies which will help shorten the gaps. In the competency gap analysis, when the parent competency is the objective competency and none of its child competencies is among the learner's competencies, then the parent competency will be included in the gap competencies, otherwise the remainder child competencies will be included in the gap competencies. The computation of gap competencies is denoted in equation. 1.

$$C_g=\{ c_{gi} \mid c_{gi} \in C_o \text{ and } c_{gi} \notin C_u \text{ and } \text{child}(c_{gi}) \notin C_u \} \cup \{ c_{gi} \mid \text{parent}(c_{gi}) \in C_o \text{ and } \exists\, \text{child}(\text{parent}(c_{gi})) \in C_u \text{ and } c_{gi} \notin C_u \} \quad (\mathbf{1})$$

Let C_u: set of learner competencies

C_o: set of objective competencies

C_g: set of gap competencies

Child(c): the child of the competency c.

Parent(c): the parent of the competency c.

All the gap competencies make up a tree (if the gap competencies construct separate trees, add a root node to connect them). Considering that when defining the concept tree and competency tree, the easy and basic nodes are always put at left side, the left-most node is apt to be the candidate recommended one. So in the target competency computation, Postorder Search algorithm could be used. If the prerequisites of the accessed point have all been acquired by the user and the accessed point has no child, then this competency will be the target competency, else its first child (the left-most node) will be the target competency. The pseudo code for the target competency computation is as following:

```
#define begin

PostSearch (ComTree, ca): process returns the next access
competency after one-step search, returns null when the search
ends. ComTree is the gap competency tree and ca is present
accessed competency.

Child(c): process returns the first child if c has child, else
returns Null.

ct: the target competency

#define end

Begin

ca=PostSearch (ComTree, Null)

do
```

```
  Cp=set of prerequisites of the concept binding to ca
  if Cp⊆Cu then
  { if (c=Child(ca) ≠null) then {ct=c, exit} ct=ca, exit}
while ((ca=PostSearch (ComTree, ca) ≠null)
ca=PostSearch (ComTree, Null)
Cp=set of the prerequisites of the concept binding to ca
ct=c | c∈Cp and c ∉Cg
End
```

3.3 Rollup Rules

The learner's competency update in the sequencing rules is controlled by rollup rules. If a competency contains sub-competency, the results of the sub-competency should be rolled up into summary result of the parent. The rollup can be controlled by rules such as:

- Satisfied if any child is satisfied;
- Satisfied if all children are satisfied;
- Satisfied if at least two children are satisfied.

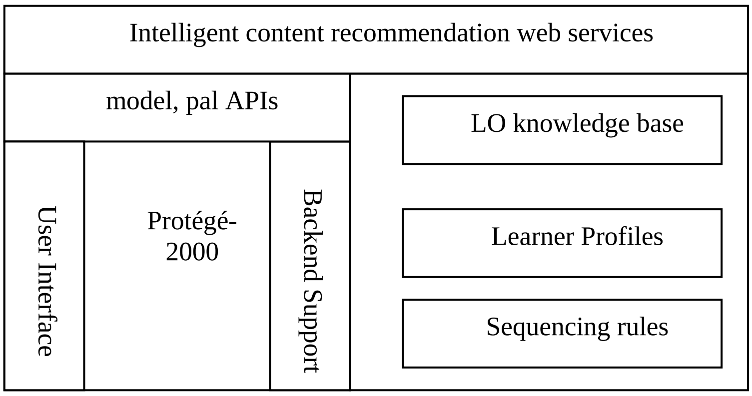
Every parent competency has rollup rules. Whenever a child competency is acquired will invoke the check-up of parent's rollup rule. If the parent competency meets, the parent competency will add to the learner's competencies and all its child competencies will be deleted.

4. Experiment

We use the ontology editor and knowledge base editor protégé-2000 (http://protégé.stanford.edu) developed by Stanford to construct the LO knowledge base. We choose protégé-2000 because it has intuitive and easy-to-use graphical user interface, has extensible plug-in architecture and has well-organized API document to help access the knowledge base. Again the Protégé-2000 has the Protégé Axiom Language (PAL) to extend knowledge modeling environment with support for writing and storing logical constraints and queries about frames in a knowledge base, which we could use to devise the sequencing rules. The protégé-2000 is programmed in Java and provides plenty of easy-to-use APIs and packages for programmers. Our application uses the edu.stanford.smi.protege.model package to access the LO knowledge base and the edu.stanford.smi.protegex.pal package to edit rules. We package the `application with .NET Framework as web services which could be accessed at different platform. Figure 2 describes the system architecture.

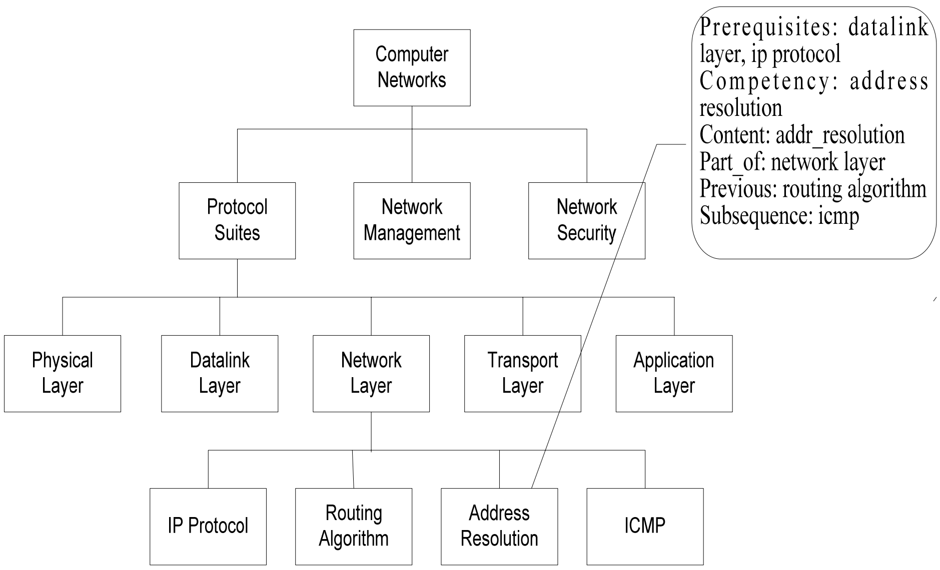
As an example in this paper, we construct a LO knowledge base of Computer Networks course. Figure 3 illustrates the directive, curricular-based and linear structure of part of the concepts of the Computer Networks. We store the concept tree in the LO knowledge base, each concept have slots such as name, competency, prerequisites, content, part_of, has_part, previous and subsequence. The competency slot that is competency.class type describes the ability/knowledge one acquires after learning the LO and passing related exercise/test. The prerequisites are also competency class type, with cardinality multiple.

Figure 2 System Architecture



Content slot stands for the LOs relate to the concept. Part_of and has_part slots are inverse slots, and the same are the slots of previous and subsequence. These four slots describe the static, linear and curricular based relationship between the concepts. The competencies for each concept are defined and created very simply, has the same name with the concept name, with slots of knowledge-point and evaluation which respectively bind corresponding concept and exercise/test. The LOs are web contents, and are added to the knowledge base together with their metadata.

Figure 3 Part of the Concept Tree of Computer Networks Course



The class learner describes the learner profiles, having slots of name, ID, objective competencies, competencies, evaluation results, learning content and control mode. The control mode slot is used to allow the learner to select contents by oneself.

PAL provides a set of special-purpose frames to hold constraints and queries that are added to a Protégé knowledge base, respectively the: PAL-CONSTRAINT and the

PAL-QUERY classes. The PAL constraint-checking engine can be run against the knowledge base to detect frames that violate those constraints. PAL is a subnet of Knowledge Interchange Format (Logic Group, 1998) and is used for writing restrictions on existing knowledge, not for asserting new knowledge. Though, we could use PAL to design sequencing rules. In our system, the action is the same, i.e. to find the target competency and the next LO to recommend or exit. So the sequencing rules could be organized as: if PAL_CONSTRAINT then PAL_QUERY. We define sequencing rules as following:

1. Create a class named Rollup_Rules and two slots for it, one is named condition with PAL_CONSTRAINT type and the other is named type with symbol type. The action of the rules is default and realized in the application program: add the parent node to the learner competencies and delete its child nodes in the learner competencies.

2. Edit instances of Rollup_Rules and apply them to parent competencies. In this experiment, we just realize two kinds of rollup rules, and we give the type value 1, 2 respectively.

rule 1 "meets if all children meet", the value of the condition slot:

```
(defrange ?pcompetency :FRAME Competency user_competencies)
(defrange ?competency :FRAME Competency)
(exists ?pcompetency
   (forall ?competency
       (and (has_part ?pcompetency ?competency)
            (type (rollup_rules ?pcompetency) 1)
            (user_competencies %learner ?competency)))
)
```

rule 2 "meets if any child meets", the value of the condition slot:

```
(defrange ?pcompetency :FRAME Competency user_competencies)
(defrange ?competency :FRAME Competency)
(exists ?pcometency
   (exists ?competency
      (and (has_part ?pcomtetency ?competency)
           (type (rollup_rules ?pcompetency) 1)
           (user_competencies %learner ?competency)))
)
```

%learner is the global variable and stands for the specific learner.

3. Create an instance of PAL_QUERY class to compute the gap competencies. We could use the AskQueries() method of the PAL.QueryEngine Interface to get all the answers of the PAL_QUERY instance.

```
  (defrange ?competency :FRAME Competency)
  (defrange ?ocompetency :FRAME Competency objective_competencies)
  (findall ?competency
  (and (not (user_competencies %learner ?competency))
       (or (and (objective_competencies %learner ?competency)
                (not (user_competencies %learner (has_part(?competency)))))
           (exists ?ocompetency
                (and (has_part ?ocompetency ?competency)
                     (user_competencies %learner (has_part ?ocompetency))))))
)
```

4. Create a class named Sequencing_Rules and two slots for it, one is named condition with PAL_CONSTRAINT type and the other is named type with symbol type. Create a subclass of the Sequencing_Rules named Initiate_Rules.

5. Edit instances of Sequencing_Rules. We compose 4 Sequencing_Rules instances,among which the one with type 1could be used as Initiate_Rules. The actions of the rules are realized in the application program. We could use the CheckKnowledgeBase() method of the PAL.ConstraintEngine Interface to check all the PAL_CONSTRAINT instances.

rule 1 "If objective competencies met then exit else {compute target competency, present LO binding to target competency}", the value of the condition slot:

```
(defrange ?competency :FRAME Competency objective_competencies)
(forall ?competency
      (user_competencies %learner ?competency)
)
```

rule 2 "If satisfied then { update user's competencies, compute target competency, present LO binding to the target competency}", the value of the condition slot:

```
(evaluation_result %learner satisfied)
```

rule 3 "If flat then represent the current LO", the value of the condition slot:

```
(evaluation_result %learner flat)
```

rule 4 "If failed then present the LO binding to one of the prerequisite", the value of the condition slot:

```
(evaluation_result %learner failed)
```

6. Realize the actions in the program.

For the actions above, the key action is computing target competency. Because the PAL couldn't assert new knowledge and do recursion, we realize this action in the program and simplify the algorithm as following:

Postorder Search to the whole competency tree. Set the ID slot of each competency node the value of the order it is searched. Ex. for the left-most node of the tree, its ID will be set to 1. Whenever the competency tree is updated, the process is invoked to update the IDs.

Compute the gap competencies checking the PAL_QUERY instance

Compute the target competency as the algorithm explained in Section3.2, the difference is that whenever invoke the PostSearch method, just simply find the next smallest ID number of the competency instead.

I tested this simple recommendation system in my class in the Network Education College (www.nec.sjtu.edu.cn) during the course review period. Finally I did a survey about the system. 158 out of the 219 students of my class turned in their feedbacks, most of whom gave me positive remarks and many helpful advices such as

> "I like the system, it helps me locate the contents that I should spent more time on quickly",
>
> "The exercises are too monotonous with only one selection type"

and

> "Each knowledge point has only one content, it is preferable that each has multiple contents with different difficulties" etc.

Thanks for all my students' tests and advices.

5. Conclusion and future work

In this paper, we give a detail description of the rule-based recommendation mechanism based on the competency gap analysis, concept tree and LO knowledge base. We test the mechanism with the knowledge base editor Protégé-2000, the PAL and the java program extending the Protégé-2000. The advantages of the system are: zero startup time and machine learning time, and the adequate content granularity---LO makes the adaptive and individualized content recommendation possible.

The method and system introduced in this paper base on simple sequencing rules (only 4 kinds of rules) that still couldn't realize full intelligent recommendation. And we just consider simple concept graph---concept tree which reflects simple relationships between concepts. The Competency class hasn't any standard description, and just serves for internal computation. It's a pity that PAL can't assert new knowledge, and other knowledge expression and inference language such as CLIPS should be considered to replace the PAL to denote the rules. How to provide auxiliary materials for the LO? Is there any better Problem Solving Method that could make the system more perfect and be reused easily by other applications? There are still many aspects to consider and improve of our system in the future.

References

Charalampos, K., Demetrios, S. and Fabrizio, C. (2001) 'An architecture for defining re-usable adaptive educational content', IEEE International Conference on Advanced Learning Technologies (ICAL'01).

Dodds, P. (2004) 'Sharable content object reference model (SCORMTM) version1.3', The SCORM Content Aggregation Model, January 30, www.adlnet.org.

Frakes, W. and Baeza-Yates, R. (1992) Information Retrieval: Data Structures and Algorithms, Prentice Hall, ISBN 0-13-463837-9.

http://protégé.stanford.edu.

http://www.celtsc.edu.cn/download/CELTS-3.1 (CD1.6).zip.

http://www.imsproject.org/simplesequencing.

Karagiannidis, C., Sampson, D. and Cardinali, F. (2001) 'Integrating adaptive educational content into different courses and curricula', Educational Technology and Society, Vol. 4, No. 3.

Li, J.Z. and Close, R. (2000) The Promise of eLearning and the Practice of Knowledge System Design, http://www.leadingway.com/eLearning.pdf.

Resnick, P. and Varian, H.R. (1997) 'Recommender systems', Communications of the ACM, March, Vol. 40, No. 3.

Salton, G. (1989) Automatic Text Processing, Addisom-wesley, ISBN 0-201-12227-8.

Logic Group (1998) Computer Science Department, Stanford University, Knowledge Interchange Format, Draft proposed American National Standard (dpANS), NCITS.T2/98-004, available online at http://logic.stanford.edu/kif/dpans.html.

Zeng, C., Xing, C.X. and Zhou, L.Z. (2002) 'A survey of personalization technology', Journal of Software, 1952~1961 (in Chinese with English abstract), Vol. 13, No. 10.