Ontology-Based Concept Map Assistant Learning System with Rule-Based Reasoning Mechanism

Kuo-Kuang Chu and Chien-I Lee

***Abstract*—Although there were some ontology-based learning researches, most of researches were off-line ontology knowledge base or without reasoning mechanism to assist user's learning. We integrated several APIs to create an ontology-based assistant learning system (ORALS) that provides online editing ontology and rules in web pages and it is compatible with these popular ontology editors. To simplify users creating ontology processes, That provides users search concept and reasoning items, all searching or reasoning results will be listed and shown by visualize concept maps. By three teachers' experiences, they applied ontology engineering technology to extract knowledge to ontology and build course concept maps and reasoning rules. There were three classes grade 7th of junior high school total 95 students joined our study with the nervous system course unit of Science and Technology. We hope this paper not only can present applied ontology to assist learning scenarios, but also help the other fields to induced ontology technologies easily.**

***Index Terms*—Ontology, e-learning, concept map, rule-based reasoning.**

I. Introduction

In the last decade, a concept map as facilitative tools to aid learning has been a dramatic increase in the number of publication. Ontology can be served as a structured knowledge representation scheme, which can assist the construction of personalized learning path. Therefore, Chen [1] proposed a novel genetic-based curriculum sequencing scheme based on a generated ontology-based concept map, Chi [2]stated that ontology technologies enable the representation of conceptual relationships among learning materials and thus ontology can serve as a structured knowledge scheme that assists in the construction of a personalized learning path. Furthermore, Chu and Lee [3] proposed an ontology-based concept map learning system with semantic rules to recommend the learning path for learners, and Lee *et al*. [4] evaluated the learning achievement difference between traditional concept map and ontology-based learning system. However, few systems have integrated web interface on rule editing and real-time concepts reasoning presentation. Bahar, Johnstone & Hansell [5] found out that over 10% university students aware that the nervous system is one of the most hard to learn units in biology, because of too many new terms meaning and lots knowledge usually be confused between similar nouns and functions. In this study, we induced the ontology reasoning on the nervous system course unit of the subject "Science and Technology" on the 7th grade junior high school. All reasoning rules is extracted and predefined by three teachers from their teaching experience and the most making mistakes from students learning. This study aims to integrated rule based inference engine to create a web concept map assistant learning system. We expect that the system not only can be applied on learning, but also help the other fields to induced ontology technologies easily.

This paper have five sections, Section I discusses about the importance of ontology-based concept map assistant learning system, the Section II reviews about concept map and ontology-based learning researches, and the Section III describes rules definition of the ontology concept map reasoning system, and Section IV is presents the system implementation and explanation, and Section V concludes on features and ongoing researches of our learning system.

Manuscript received April 10, 2013; revised June 20, 2013.

The authors are with National University of Tainan, Tainan, Taiwan (e-mail: chukk@tn.edu.tw, leeci@mail.nutn.edu.tw).

II. Literature Review

A. Concept Map

Concept maps are graphical representations of knowledge that consist of concepts and the relationships among them, as in Fig. 1. As they draw concept maps, students reflect upon and construct their own knowledge structure after learning: the concept map providing a visual representation of the students' knowledge of a specific topic. Puntambekar *et al.* [6] found that students in a class using concept maps visited more goal-related concepts and spent more time on them, improved more on an essay question, and performed better when tested on their depth of knowledge. Concept mapping allows students to visualize for themselves their knowledge and growing understanding. The process of building a concept map can itself be a valuable learning experience, helping students focus on the relationships between concepts in the domain knowledge.

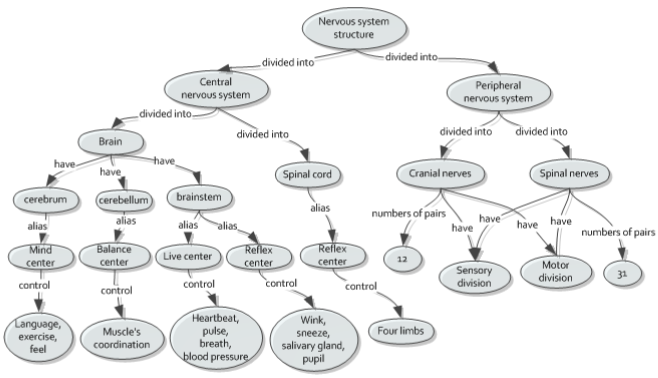

Fig. 1. Hierarchical concept map of nervous system structure.

DOI: 10.7763/IJEEEE.2013.V3.221

B. Ontology-Based Learning

Noy & McGuinnes [7] explains that why use ontologies for some reasons:

- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To make domain assumptions explicit
- To separate domain knowledge from the operational knowledge
- To analyze domain knowledge

Abel *et al.* [8] proposed an ontology building process an Ontology-based Organizational Memory for e-learning. Henze *et al.* [9] provided a framework for personalized e-Learning in the semantic web. Fisher [10] suggested to establish domain ontology, and to perform pedagogy in the way of describing ontology knowledge, then to assist learners to understand digestive system. Fisher points out that the digestive system could use the way of ontology to describe the relationship between concepts. Several studies [3], [4], [11], [12] generated different ontology-based concept map e-learning systems and provided findings and suggestions to facilitate personalized learning. General speaking, ontology is a key technology that enables the semantic web feasible, because of that hybrids machine readable and human understanding symbols.

C. Ontology for Reasoning

The inference engine is the key role for ontology-based knowledge reasoning. There are some inference engines that can use in the reasoning process. For example, RacerPro [13], Jess (A Java Expert System Shell) [14] and Apache Jena [15]. Therefore, only the Apache Jena is a free charge inference engine, so we choose it for our research project. Jena provides a collection of tools and Java libraries to develop semantic web, linked-data apps and tools.

One of the main reasons for building an ontology-based assistant learning system is to use a reasoner to derive additional rules about concept maps. For example, the assertion "Fred is a Fish" entails the deduction "Fred is an Animal". There are many different styles of automated reasoner, and very many different reasoning algorithms. Jena includes support for a variety of reasoners through the inference API. A common feature of Jena reasoners is that they create a new RDF model which appears to contain the triples that are derived from reasoning, as well as the triples that were asserted in the base model. This extended model is, nevertheless, still conforms to the contract for Jena models. So it can be used wherever a base model can be used. The ontology API exploits this feature: the convenience methods the ontology API provides can query an extended inference model in just the same way as a plain RDF model.

Ontology reasoning will be transformed to the triples format to process by Jena API, so we must create reasoning rules in Triple format. There is an example complete rule file which includes the RDFS rules and defines a single extra rule. The definition of reasoning rules are listed as below:

Rule allID illustrates the function use for collecting the components of an OWL restriction into a single data structure which can then fire further rules. Rule all2 illustrates a forward rule which creates a new backward rule and also calls the print procedural primitive. Rule max1 illustrates use of numeric literals.

```
# Example rule file
@prefix pre: <http://jena.hpl.hp.com/prefix#>.
@include <RDFS>.
[rule1: (?f pre:father ?a) (?u pre:brother ?f) ->
        (?u pre:uncle ?a)]
```
(1)

```
[allID: (?C rdf:type owl:Restriction),
   (?C owl:onProperty ?P), (?C owl:allValuesFrom ?D)  -> (?C
   owl:equivalentClass all(?P, ?D)) ]
[all2: (?C rdfs:subClassOFf all(?P, ?D)) -> print('Rule for ', ?C)
[all1b: (?Y rdf:type ?D) <- (?X ?P ?Y), (?X rdf:type ?C) ] ]
[max1: (?A rdf:type max(?P, 1)), (?A ?P ?B), (?A ?P ?C)
-> (?B owl:sameAs ?C) ]
```
(2)

III. Ontology Reasoning System

A. Definition of Reasoning Rules

[rule1:(?a rdf:type fa:behavior)(?a fa:passing to fa:brain)->(?a owl:Class fa:mind control behavior)] (3)

Rule1: if exists an *instance* a is the class *behavior*, and a fact *passing to fact:brain* that belongs to the class *from mind control behavior*.

[rule2:(?a rdf:type fa:behavior)(?a fa:passing to fa:brainstem) ->(?a owl:Class fa:reflex control behavior)] (4)

Rule2: if exists an *instance a* that is the class *behavior*, and if a is the *passing to brainstem* that derived it belongs to the class *reflex control behavior*.

[rule3:(?a rdf:type fa:behavior)(?a fa:passing to fa:to spinal cord)->(?a owl:Class fa:reflex control behavior by)] (5)

Rule3: if exists an *instance a* that is the class *behavior*, and if a is *passing to spinal cord* that derived it belongs by the class *reflex control behavior by*.

[rule4:(?a rdf:type fa:organ)(?a fa:control ?b)(?b rdf:type fa:behavior)(?b fa:passing to fa:brain)->(?a owl:Class fa:mind control organ)] (6)

Rule4: if exists an *instance* a that is the class *organ*, and if a is able to control the class *behavior* b, and b is *passing to brain* that derived it belongs to the class *organ*.

[rule5:(?a rdf:type fa:organ)(?a fa:control ?b)(?b rdf:type fa:behavior)(?b fa:passing to fa:brainstem)->(?a owl:Class fa:reflex control organ)] (7)

Rule5: if exists an *instance a* that is the class *organ*, and if a is able to *control* b, and b *is passing to brainstem* that derived it belongs to the class *reflex control organ.*

[rule6:(?a rdf:type fa:organ)(?a fa:control ?b)(?b rdf:type fa:behavior)(?b fa:passing to fa:to spinal cord)->(?a owl:Class fa:reflex control organ)] (8)

Rule6: if exists an *instance a* that is the class *organ*, and if a is able to *control* b, and b *is passing to spinal cord* that derived it belongs to the class *reflex control organ*.

[rule7:(?a rdf:type fa:event)(?a fa:action fa:brain)->(?a owl:Class fa:mind control by event)] (9)

Rule7: if exists an *instance a* that is the class *event*, and if a is the *action brain* that derived it belongs to the class *mind control by event.*

[rule8:(?a rdf:type fa:event)(?a fa:action fa:brain to spinal cord)->(?a owl:Class fa:mind control by event)] (10)

Rule8: if exists an *instance a* that is the class *event*, and if a is the *action brain to spinal cord* that derived it belongs to the class *mind control by event.*

[rule9:(?a rdf:type fa:event)(?a fa:action fa:spinal cordbrain)->(?a owl:Class fa:mind control by event)] (11)

Rule9: if exists an *instance a* that is the class *event*, and if a is the *action spinal cordbrain* that derived it belongs to the class *mind control by event.*

[rule10:(?a rdf:type fa:event)(?a fa:action fa:spinal cordbrain to spinal cord)->(?a owl:Class fa:mind control by event)] (12)

Rule10: if exists an *instance* a that is the class *event*, and if a is the *action spinal cordbrain to spinal cord* that derived it belongs to the class *mind control by event.*

[rule11:(?a rdf:type fa:event)(?a fa:action fa:brainstem)->(?a owl:Class fa:reflex control by event)] (13)

Rule11: if exists an *instance* a that is the class *event*, and if a is the *action brainstem* that derived it belongs the class *reflex control by event*.

[rule12:(?a rdf:type fa:event)(?a fa:action fa:至spinal cord)->(?a owl:Class fa:reflex control by event)] (14)

Rule12: if exists an *instance a* that is the class *event*, and if a is the *action to spinal cord* that derived it belongs to the class *eflex control by event*.

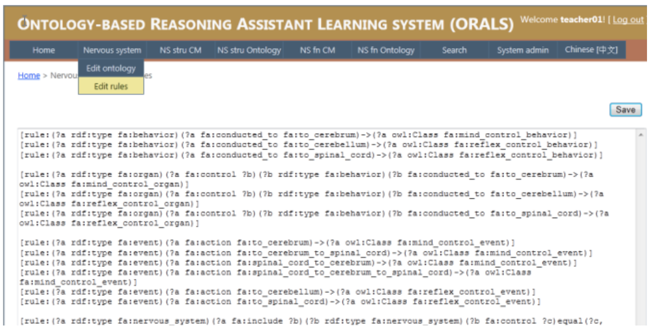

Fig. 2. Edit rules in learning system.

Fig. 2 shows rules editing web page of the Ontology-Based Reasoning Assistant Learning System (ORALS) for users.

B. Inference Engine

Jena OWL reasoner provides forward, backward and hybrid reasoning approaches, Fig. 3 illustrates these data flows and relationships of inference engines.

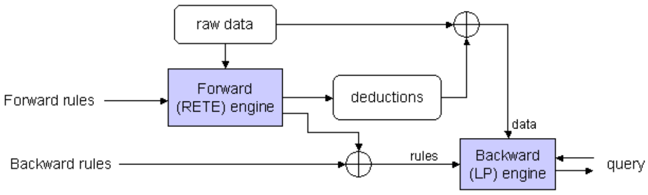

Fig. 3. Jena inference engine diagram [15].

C. Query Ontology

When we need to search the ontology, this system will transform to SPARQL [16] language to apply the query in Jena API. There is an example to demonstrate how to search these properties in the specific class as Fig. 4.

```
String queryString1 = "PREFIX fa: <" + NS + "> " +
"Prefix geodi:<http://www.owl-ontologies.com/geodi.owl#> " +
"Prefix amo:<http://cmrc.ucc.ie/ontologies/org/esri/amo.owl#> " +
"Prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> " +
"Prefix owl:<http://www.w3.org/2002/07/owl#> " +
"Prefix list:<http://Jena.hpl.hp.com/ARQ/list#> " +
" SELECT ?m " +
" WHERE { ?m rdfs:" + gettype + " fa:" + className + " }";        (15)
```

Fig. 4. Query example by using SPARQL to search in Jena.

```
digraph g {rankdir=LR;graph [bgcolor="#E4F0F0"]
node[shape=box, style=filled,];to_brainstem;
node[style=solid, color="#0C54A8",fontcolor="#663333"];
subgraph cluster0 { style=solid; color="#600000";
node [ shape=ellipse, color="#CF354E"];nerves; center_conduct;
label = "type";}
to_brainstem -> nerves
to_brainstem -> center_conduct
hiccup -> to_brainstem [label="conducted_to"];
vomit -> to_brainstem [label="conducted_to"];
sneeze -> to_brainstem [label="conducted_to"];
cough -> to_brainstem [label="conducted_to"];
maintain_life -> to_brainstem [fontsize=11, fontcolor=Brown,
label="conducted_to"];
salivates_when_smell_food -> to_brainstem [label="action"];
unconscious_wink -> to_brainstem [label="conducted_to"];
salivate -> to_brainstem [label="conducted_to"];label =
"to_brainstem relevant concepts"}        (16)
```

Fig. 5. Graphviz gv format code.

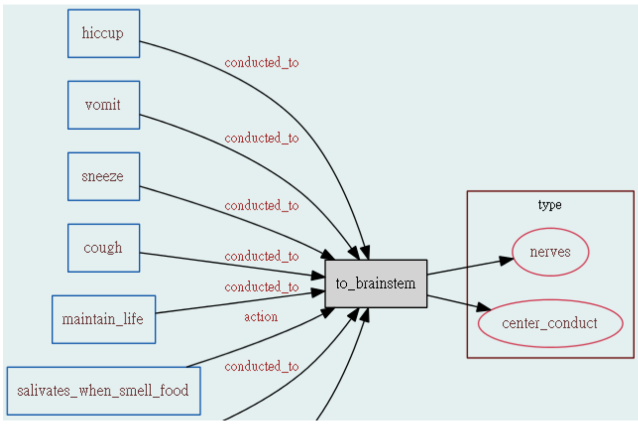

Fig. 6. Draw the Concept map by Graphviz.

D. Visualize Concept Map

Jena API provides the textual ontology query result, we improved it became the visualize concept map with Graphviz 2.28 [17]. The system would convert the query result to gv format file as Fig. 5, and import to the Graphviz to generate the visualize concept map as Fig. 6.

IV. System Implementation

A. System Modules Architecture

According above section kernel technology, we implemented the Ontology-Based Reasoning Assistant Learning System (ORALS) that have five modules as Fig. 7.

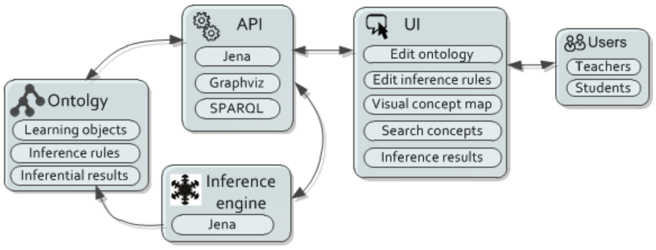

Fig. 7. System modules architecture diagram.

1) Ontology: it contains learning objects that stored in OWL lite file and text-plain inference rules file. When users search the inference items, all inferential results will convert to Graphviz gv file and store it.
2) API: we use Jena, Graphviz and SPARQL API library as the framework to implement the system.
3) Inference engine: we use the generic Jena OWL reasoner as rules inference engine.
4) User Interface: ORALS provides web-based interface to create and edit ontology, and the inference rules can be edit and stored by the web page of system. All search results, including concept map and inference items, will be draw and present in the web page.
5) Users: the system provides teacher and student account to logon the learning platform, all users' action will be record in the logs.

B. System Features

The data flows of ORALS are major from the user interfaces. The system offers student and teacher that has different permission to access system. When logged on users submit a request, the system will call API libraries to retrieve ontology and rules, and drive the inference engine to generate inference items if it is necessary.

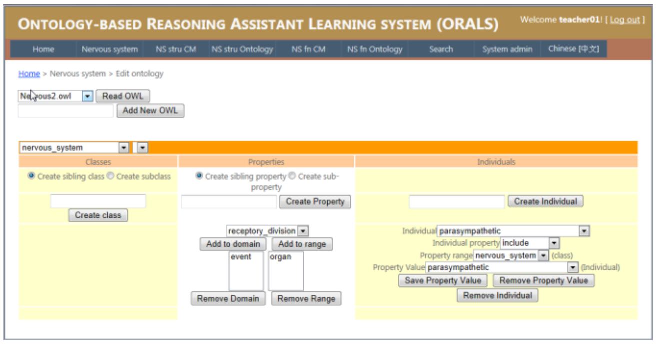

Fig. 8. Edit ontology.

Protégé [18] is a popular ontology editor and knowledge acquisition system, we can use it to build course ontology. But the build-in inference engine is too simple, it is only to check inconsistency, so it must plug-in the other inference engine, like as RacePro, Jess, to inference common rules. In this study, we design the ontology edit web interface as Fig. 8 to replace the Protégé, therefore it is also compatible with Protégé. In this ontology editor, we can create classes, properties and individuals, and assign the domain and range of the property. And then we can create individuals one by one, and assign detail individual property values.

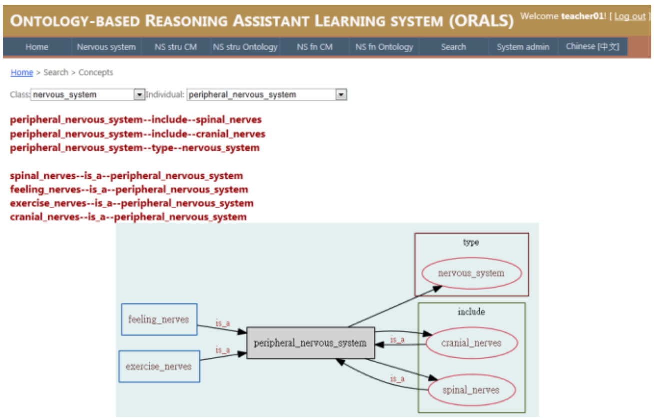

Fig. 9. Search concepts.

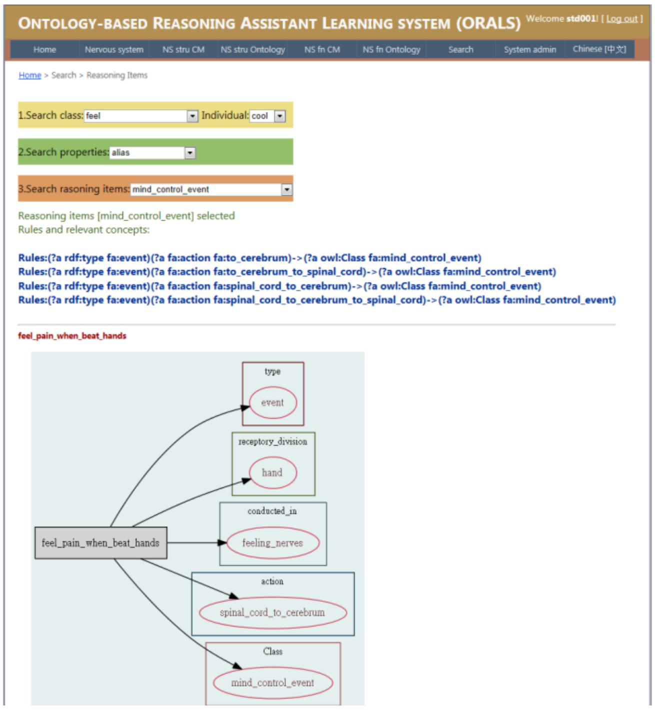

Fig. 10. Search reasoning items.

ORALS provides ontology search concepts and rules reasoning items:

1) Search concepts: users can select specific class and individual to filter relent concept maps, it illustrated as Fig. 9, and users can also select specific property to search concepts, all match concepts will be listed in red font color and drawn with concept maps.
2) Search rules reasoning items: users can select class, individual and property that will be inferred by Jena inference engine with rules. All match rules will be listed in blue font color, and the match concepts drawn with concept maps as Fig. 10.

C. System Development Technology

The Ontology-based reasoning assistant learning system (ORALS) integrated many technologies to achieve the goal of this study, Table I is listing and describing these technologies.

TABLE I: THE LIST OF SYSTEM DEVELOPMENT TECHNOLOGY

Category	Description
Ontology store	OWL lite
Ontology API	Jena .NET[19] (based on Jena 2.6.2)
Visualize library	Graphviz 2.28
Ontology editor	Built in web-based editor
Inference engine	Jena 2 OWL reasoner
Language	C#、OWL、RDF、SPARQL
Development tool	Microsoft Visual Studio 2010
System framework	.NET Framework 4.0

V. CONCLUSION

In this study, we integrated Jena, Graphviz and SPARQL API to create an ontology-based assistant learning system (ORALS), and induced the nervous system course unit of grade 7th Science and Technology in ontology engineering to establish ontology knowledge base and reasoning rules. There are 3 classes of 3 junior high schools total 95 students joined to this study. In order to simplify users creating ontology processes, ORALS offers editing ontology and rules web pages that is compatible with these popular ontology editors. The system provides users search concept and reasoning items, all searching or reasoning results will be listed and shown by visualize concept maps.

To assess the usability and reliability of the ORALS, three classes' students of three junior high schools had attended the experiment, we also want to study if students' performance have any significant difference between using the ORALS and traditional learning with concept map approach. All data now are going processing and analyzing, we hope that the research results will finish and publish lately, and expect that it is not only using about learning, but also can help the other fields to induced ontology technologies easily.

REFERENCES

[1] C. M. Chen, "Ontology - based concept map for planning a personalised learning path," *British Journal of Educational Technology,* vol. 40, no. 6, pp. 1028-1058, 2008.

[2] Y. L. Chi, "Ontology-based curriculum content sequencing system with semantic rules," *Expert Systems with Applications,* vol. 36, no. 4, pp. 7838-7847, 2009.

[3] K. K. Chu, C. I. Lee, and R. S. Tsai, "Ontology technology to assist learners' navigation in the concept map learning system," *Expert Systems with Applications,* vol. 38, no. 9, pp. 11293-11299, Sep, 2011.

[4] C. I. Lee, Y. F. Yang, and J. I. Xiao, "To enhance the learning achievement of students by establishing an assistant-learning system based on the knowledge ontology," in *Taiwan e-Learning and Digital Archives Program (TELDAP) International Conference,* Taipei, Taiwan, 2011.

[5] M. Bahar, A. Johnstone, and M. Hansell, "Revisiting learning difficulties in biology," *Journal of Biological Education,* vol. 33, no. 2, pp. 84-86, 1999.

[6] S. Puntambekar, A. Stylianou, and R. Hübscher, "Improving navigation and learning in hypertext environments with navigable concept maps," *Human-Computer Interaction,* vol. 18, no. 4, pp. 395-428, 2003.

[7] N. F. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology," *Stanford knowledge systems laboratory technical report KSL-01-05 and Stanford medical informatics technical report SMI-2001-0880*, 2001.

[8] M. H. Abel, A. Benayache, D. Lenne, C. Moulin, C. Barry, and B. Chaput, "Ontology-based Organizational Memory for e-learning," *Educational technology & society,* vol. 7, no. 4, pp. 98-111, 2004.

[9] N. Henze, P. Dolog, and W. Nejdl, "Reasoning and ontologies for personalized e-learning in the semantic web," *Educational Technology & Society,* vol. 7, no. 4, pp. 82-97, 2004.

[10] K. M. Fisher, "The importance of prior knowledge in college science instruction," *Reform in Undergraduate Science Teaching for the 21st Century. Greenwich, CT: Information Age Publishing*, pp. 69-83, 2004.

[11] K. K. Chu and C. I. Lee, "Ontology-based concept map learning path reasoning system using SWRL rules," in *IASTED Technology Conferences MS2010*, Banff, Alberta, Canada 2010.

[12] K. K. Chu and C. I. Lee, "A study of transfer of learning applying ontology-based assistant learning system," *Industrial Conference on Data Mining - Poster and Industry Proceedings*, 2012, pp. 67-79.

[13] V. Haarslev, K. Hidde, R. Möller, and M. Wessel, "The RacerPro knowledge representation and reasoning system," *Semantic Web,* vol. 3, no. 3, pp. 267-277, 2012.

[14] E. J. Friedman-Hill, "Jess, the java expert system shell," *Distributed Computing Systems, Sandia National Laboratories, USA*, 1997.

[15] A. Jena. Reasoners and rule engines: Jena inference support. (2013). [Online]. Available: http://jena.apache.org/documentation/inference/

[16] E. Prud'Hommeaux and A. Seaborne, "SPARQL query language for RDF," *W3C recommendation,* vol. 15, 2008.

[17] J. Ellson, E. Gansner, L. Koutsofios, S. C. North, and G. Woodhull, "Graphviz—open source graph drawing tools." *Lecture Notes in Computer Science,* pp. 483-484.

[18] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu, "The evolution of Protégé: an environment for knowledge-based systems development," *International Journal of Human-computer studies,* vol. 58, no. 1, pp. 89-123, 2003.

[19] Jena NET. (2012). A flexible .NET port of the Jena semantic web toolkit. [Online]. Available: http://semanticweb.org/wiki/Jena_.NET

Kuo-Kuang Chu was born in Hsinchu, Taiwan, R.O.C. in 1966. He received the B.S. degree in Chemistry from the Tamkang University, Tamsui, Taiwan, in 1987, and the first MS. Degree in Chemical Engineering from the National Cheng Kung University, Tainan, Taiwan, in 1992, and the second MS. Degree in the Graduate Institute of Computer Science and Information Education, National Tainan Teachers College, Tainan, Taiwan, in 1998. Now he is PhD student in the department of information and learning technology, National University of Tainan since 2006. He currently is a teacher of public elementary school with 14 years' service in Tainan, Taiwan. His research interests include database, software engineering, ontology engineering, semantic web and e-learning and information system development. He held the leader of the Educational Network Center of Tainan, Taiwan from 2000 to 2011, and he led his team to develop the urban-based information system of Tainan City, Taiwan. And he received the MOE's Taiwan Academic Network Outstanding Contribution Award in 2003.

Chien-I Lee was born in Taipei, Taiwan, R. O. C., in 1965. He received the B.S. degree in computer science from Feng Chia University in 1987 and M.S. degree in applied mathematics from National Sun Yat-Sen University in 1993. He received the Ph.D. degree in computer science from National Chiao Tung University in June 1997 and then joined the Graduate Institute of Computer Science and Information Education, National University of Tainan, Tainan (NUTN), Taiwan. He is currently the vice professor of department of information and learning technology, NUTN. His research interests include data mining, e-learning, multimedia databases and information retrieval. He was the director of computer and network center in 2002, the library direct in 2006 and the dean of general affairs in 2007. Now he is the dean of student affairs since 2012, NUTN.