Computing and Informatics, Vol. 36, 2017, 1312–1334, doi: 10.4149/cai_2017_6_1312

SEMAG: A NOVEL SEMANTIC-AGENT LEARNING RECOMMENDATION MECHANISM FOR ENHANCING LEARNER-SYSTEM INTERACTION

Cuong Dinh Hoa Nguyen

Department of Computer Science
Faculty of Science, Khon Kaen University, Thailand
&
Department of Business Information System
College of Economics, Hue University, Viet Nam
e-mail: ndhcuong@hce.edu.vn

Ngamnij Arch-int*, Somjit Arch-int

Department of Computer Science
Faculty of Science, Khon Kaen University, Thailand
e-mail: {ngamnij, somjit}@kku.ac.th

Abstract. In this paper, we present SEMAG - a novel semantic-agent learning recommendation mechanism which utilizes the advantages of instructional Semantic Web rules and multi-agent technology, in order to build a competitive and interactive learning environment. Specifically, the recommendation-making process is contingent upon chapter-quiz results, as usual; but it also checks the students' understanding at topic-levels, through personalized questions generated instantly and dynamically by a knowledge-based algorithm. The learning space is spread to the social network, with the aim of increasing the interaction between students and the intelligent tutoring system. A field experiment was conducted in which the results indicated that the experimental group gained significant achievements, and thus it supports the use of SEMAG.

Keywords: Intelligent tutoring system, multi-agent system, personalized learning recommendation, instructional semantic web rules

Mathematics Subject Classification 2010: 68Q55, 68T30, 68U35

* corresponding author

1 INTRODUCTION

In recent years, e-learning has become a growing trend in education, due to its many advantages, especially in breaking the limitations of time and space within the traditional learning model. Many such learning management systems (LMS) have been introduced, and are used widely (e.g., Moodle[1], LAMS[2], Atutor[3], etc.). Despite the technological advantages of the LMS, most of these systems incorporate a teacher-centric design, in which curriculum, learning contents, and quizzes have been predefined by the teacher(s), and enrolled learners must work within these parameters with no consideration of their individual abilities or preferences. Several research attempts have been recently made to personalize the learning process through an LMS incorporated learner-centric design or an intelligent tutoring system (ITS) [1, 5, 8, 10].

There are different approaches to ITS, which can be categorized into two categories: the data-mining approach, and the knowledge-based approach. Both of them share the common target of improving learners' performances by providing recommendations generated by their knowledge models. Data-mining approaches depend mainly on the learner's learning data (e.g., scores, behavior logs, etc.) where building their knowledge models; whereas the knowledge-based approach utilizes the teacher's knowledge and experiences, which are valuable resources that assist in providing efficient learning recommendations, and improve the learners' performances. As a result, numerous knowledge-based ITS studies have been introduced in the literature, especially those which have utilized the advantages of knowledge representation of semantic web technologies in various aspects, such as domain modeling [3, 12], educative curriculum management [9], learning assessment [11], or learning recommendation [26].

Although there are many approaches to ITS the process of providing recommendations generated by their knowledge models can be generalized as a three-step process. Firstly, ITS generated recommendations are sent to relevant learners. Secondly, learners, having received the recommendations, are encouraged to study them independently. Lastly, ITS checks the efficiency of the recommendations, by asking learners to join quizzes. This working scenario is capable of providing efficient recommendations but is limited in its ability to positively motivate the learners' learning attitudes. That is because the ITS does not create a competitive learning environment which can encourage learners to improve their performances through their own self-study. Therefore, building a competitive learning environment attracting the learners to openly participate in the learning process through various learner-ITS interaction scenarios is a rising research trend within the ITS domain.

In the view of system development, most ITS models were developed as standalone systems for specific subjects, and did not utilize the advantages of the avail-

[1] `https://moodle.org/`
[2] `http://lamsfoundation.org/`
[3] `http://atutor.ca/`

able LMS (e.g. Moodle). In this context, difficulties were encountered in deploying ITS for popular use. Additionally, virtual space learners simultaneously used both e-learning systems, as well as social networks. In fact, learners who spend greater amounts of time in social networks often have lower grade point averages [14]. However, studies have shown that students are open-minded to the concept of using social networks for educational purposes [4, 21]. Therefore, social networks may be integrated into e-learning systems in order to broaden the learning space, and increase the learners' ITS interaction, with the aim of enhancing learning performances.

Based on the aforementioned analyses, our research objectives focus on solving the following two problems: first, to build a competitive learning environment based on learner-ITS interaction, delivering personalized learning recommendations instantly and semantically; and second, to present a method of ITS construction which satisfies the requirements of a broadened learning space in social networks, in order to increase interaction with learners, integrate available LMS, and utilize its technical advantages as a valuable learning resource.

In this study, in order to reach the research objectives, we present SEMAG – a novel semantic-agent learning recommendation mechanism which is the core component necessary to generate personalized learning recommendations and to create a competitive and interactive learning environment. Specifically, SEMAG uses instructional semantic web rules to develop personalized recommendations while a multi-agent system (MAS) is constructed to deal with the workload of dynamic requirements in the learning environment and to deliver recommendations instantly. Based on the semantic-agent reasoning mechanism, two algorithms are proposed in order to create the desired competitive and interactive learning environment. Furthermore, SEMAG is integrated into both the LMS and social network for the purposes of utilizing LMS facilities and widening the learning space. A field experiment, conducted in the College of Economics, Hue University, Viet Nam, showed promising results which support the use of SEMAG.

The rest of this paper is structured as follows: Section 2 presents the SEMAG framework, including system architecture, domain ontology, instructional semantic web rules, and reasoning algorithms for generating personalized learning recommendations. In Section 3, we outline the experiments, and discuss the results. We review state-of-the-art ITS studies and compare them with the proposed SEMAG framework in Section 4. And lastly, Section 5 presents our conclusions and recommendations for future work.

2 SEMAG FRAMEWORK

Figure 1 shows the overall architecture of the SEMAG framework that is comprised of three main components: the LMS, the back-end MAS, and the social network. Within this framework, the LMS is the major e-learning environment where all e-learning activities occur. The MAS is responsible for:

1. monitoring the students' learning activities and results;
2. recognizing the students' learning contexts;
3. delivering personalized learning recommendations to each student; and
4. increasing students' ITS interaction, via personalized recommendation messages and questions.

The social network is used as a means to increase interaction and maintain contact with students in a virtual society.

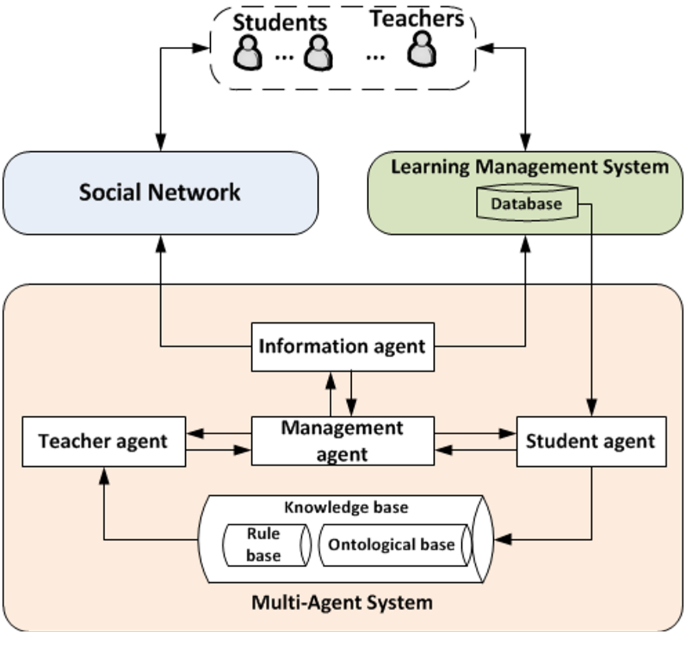


Figure 1. SEMAG framework

Specifically, the MAS has four kinds of agents: the Management agent, Student agent, Teacher agent, and Information agent; where each agent has its own functionality. The Management agent is in charge of controlling the interaction process between agents, and overseeing the agent society, by creating, adding, or removing the agents. The Student agent monitors the learning process, as well as student activities, and their results; and it is also responsible for retrieving the learners' data in the LMS database, and updating the knowledge base of the MAS. The Teacher agent plays the obvious role of 'teacher' in the e-learning environment. This agent implements reasoning to generate personalized learning recommendations or questions for each student. Lastly, the Information agent is in charge of delivering personalized messages generated by the Teacher agent to social network and the LMS. The recommendation messages are built from natural language templates.

The cyclic process of the MAS is described as follows: At start up, the Management agent is created with one living instance, and lives infinitely. Other agents are then created, added, or removed by the Management agent; but at any time, there is at least one agent per the kinds of lives. Next, the Management agent requires Student agent(s) to retrieve data from the LMS database and update the MAS knowledge base. After that, the Management agent compels the Teacher agent(s) to implement reasoning, in order to generate personalized recommendations or personalized questions. The Teacher agent(s) then returns the inference results to the Management agent, which transfers the recommendations received from the Teacher agent(s) to the Information agent(s). In the last phase of the MAS process, the Information agent(s) builds recommendation messages in natural language, and delivers these messages to the LMS and social network.

2.1 Domain Ontology

Because SEMAG is a knowledge-based approach, utilizing semantic web technologies, ontology plays an important role in representing domain knowledge. In this work, we developed an ontology for the purpose of specifying the domain knowledge of the undergraduate-level subjects, in our public university, which describes the user profiles of both students and teachers. Within the domain knowledge, the ontology describes the learning materials, learning sequences, and the structure of the credit-based courses. The concepts, relationships, and the instances of the selected subject are also specified. The domain ontology also captures the user profiles, personal information, teaching activities (for teachers), and learning activities and results (for students). Furthermore, under university regulations, all of the learning data must be capable of integration into the university's educative information system. Therefore, the ontology also covers aspects relating to the credit-based education program of our university, such as the enrollment regulations, constraints of learning process, and so on.

In order to build the domain ontology, we adopted the popular method of Noy and McGuiness [19] which is a collaborative effort on the part of the university's lecturers, administrative staff and the knowledge engineers. The developmental process involves determining the scope of the domain ontology, enumerating all related terms in the ontology, and defining the concepts and concept hierarchy. Next, the properties (or relations) including object properties and data properties were determined. Lastly, we specify the constraints for each relation. This developmental process is repeated until all of the participants reach a consensus concerning the resulting ontology.

We developed the domain ontology using Protégé[4], which is open-source software for editing and managing ontologies. The domain ontology was expressed in

[4] `http://protege.stanford.edu/`

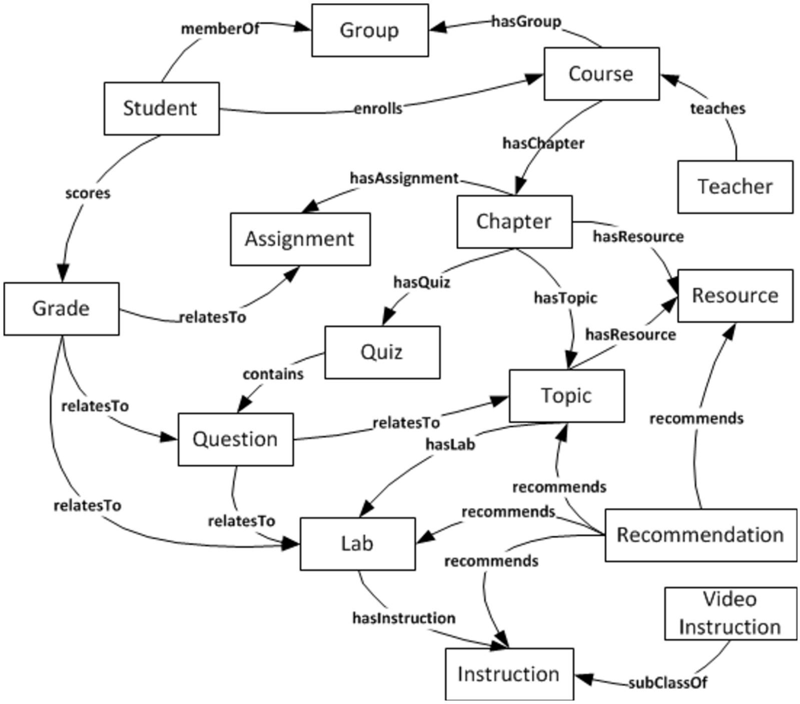


Figure 2. An excerpt of the domain ontology

OWL[5] language, and its consistency was checked by Fact++[6] reasoner. This ontology had 37 concepts, 65 object properties, 42 data properties, and 152 constraints (78 exact cardinality, 40 maximum cardinality, and 34 minimum cardinality constraints). Figure 2 shows an excerpt of the domain ontology. In general, the domain ontology can be defined as in Definition 1.

Definition 1 (Domain ontology)**.** The domain ontology is a triple $O = \langle C, R, I\rangle$ where C is the set of concepts, R is the set of relations (or properties), and I is the set of instances.

2.2 Transferring Learning Instructions to Instructional Semantic Web Rules

While domain ontology plays a significant role in the knowledge-specification foundation, its major limitation falls within the capture of implicit relations. Therefore, the rule-based approach is often applied to find implicit relations, which cannot be

[5] `http://www.w3.org/TR/owl-features/`
[6] `http://code.google.com/p/factplusplus/`

expressed solely through an ontological reasoning process. On the one hand, ontologies require rules with which to derive further information that cannot be initially captured. On the other hand, the rules employ the ontological concepts, relations, and instances in order to perform their reasoning tasks. Many researches have recently expressed learning instructions under the form of the semantic web rules, which may be referred to as instructional semantic web rules [6, 26, 27].

Most of the rule-based solutions used SWRL [13] to define rules [15]. SWRL rules are advantageous in their simplicity, and are supported by an intermediate inference engine; however, they are severely limited in their expressiveness capacity. Recent studies have attempted to overcome this limitation, by defining the rules through the use of SPARQL[7] language [15, 17, 24]. SPARQL is more expressive than SWRL, because it enables features like UNION and FILTER clauses. Additionally, SPARQL-based rules do not require an intermediate inference engine, and therefore do not require the introduction of further communication overhead. A rule acts as a form of implication between an antecedent and a consequent, which suggests that if the antecedent holds, or is "true", the consequent also holds. Within SPARQL-based rules, CONSTRUCT queries are often used to express rules in combination with WHERE and/or FILTER clauses. The antecedent is presented in WHERE clause which aims at matching graph patterns, while the constraints are often expressed in FILTER clause(s). The consequence is then placed in a CONSTRUCT clause, responsible for establishing new relations. In this study, we have selected the CONSTRUCT query of the SPARQL language to build the instructional semantic web rules.

In order to build the instructional semantic web rule-base of learning instructions, we propose the following three-step process.

1. Interview lecturers. The knowledge engineer interviews the lecturers about their learning instructions and teaching experiences, what helps learners to overcome their learning difficulties; that, in turn, improves the learners' practical skills and broaden their knowledge.

2. Write rules in natural language. Based on the interview results, we can determine the instructional rules. The rule contents are rewritten in natural language. Steps 1. and 2. are repeated until all of the participants (lecturers and knowledge engineers) reach a consensus.

3. Build CONSTRUCT-based rules. Based on the instructional rules of the natural language, we build instructional semantic web rules, following the SPARQL language CONSTRUCT query format.

The above three-step process was applied to build the instructional semantic web rule-base for the Networking subject within our test university. Because Networking requires that students understand theory, and have fluent practical skills; the instructional rules must satisfy the above two requirements. We have therefore

[7] `http://www.w3.org/TR/rdf-sparql-query/`

divided our rules into two groups: theory-oriented rules and practice-oriented rules. The result rule-base consisted of 58 instructional semantic web rules (22 rules for theoretical recommendations, and 36 rules for practical recommendations). For example, a CONSTRUCT query of the instructional rule *"If a student whose current performance is B cannot pass a difficult question of a networking experiment (or Lab), then a related video instruction and a related difficult learning resource are introduced to that student"* is shown in Listing 1. Students' learning performances are ranked (A, B, C, D, and F) based on their total scores. The learning resources, including questions within the question bank, are assigned difficulty levels by the lecturers (*very easy, easy, quite difficult, difficult*, or *very difficult*).

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX mydef: <http://www.hce.edu.vn/intellearn.owl#>
CONSTRUCT{
       ?rec mydef:recommends ?v;
            mydef:recommends ?res;
            mydef:recommends ?l .
            ?s mydef:hasRec ?rec .
       ?g mydef:isRecommended 1 .}
WHERE{
       ?s rdf:type mydef:Student.
       ?s mydef:hasID "{studentID}"^^xsd:string.
       ?s mydef:hasPerformance "B"^^xsd:string.
       {SELECT ?g ?q
              WHERE{
                     ?g rdf:type mydef:Grade.
                     ?q rdf:type mydef:Question.
                     ?q mydef:hasDifficultLevel
                        "difficult"^^xsd:string.
                     ?g mydef:relatesTo ?q.
                     ?g mydef:hasGrade ?hg.
                     BIND (xsd:float(?hg) as ?hg_value).
                     ?q mydef:hasDefaultGrade ?dfg.
                     BIND (xsd:float(?dfg) as ?dfg_value).
                     FILTER(?hg_value < ?dfg_value).
                     FILTER NOT EXISTS{?g mydef:isRecommended ?g1}
              } LIMIT 1}
       ?s mydef:scores ?g.
       ?l rdf:type mydef:Lab.
       ?q mydef:relatesTo ?l.
       {SELECT ?v
              {SELECT (SAMPLE(?video) AS ?v)
                     WHERE{?video rdf:type mydef:VideoInstruction.
```

```
35                                ?l mydef:hasInstruction ?video.}}}
36          {SELECT ?res
37                {SELECT (SAMPLE(?resource) AS ?res)
38                       WHERE{?resource rdf:type mydef:Resource.
39                              ?resource mydef:hasDifficultLevel
                                   "difficult"^^xsd:string.
40                              ?resource mydef:relatesTo ?l.}}}
41         {SELECT ?rec
42                WHERE{ ?rec rdf:type mydef:Recommendation.
43                       FILTER NOT EXISTS{?rec mydef:recommends ?r1}
44                } LIMIT 1}
45  }
```

Listing 1. An example of CONSTRUCT-based rule

- Lines 7–11 express new relations resulting from the rule's reasoning process. These relations involve the instances (*?rec, ?s, ?v, ?res, ?l*, and *?g*) of the concepts *Recommendation*, *Student*, *VideoInstruction*, *Resource*, *Lab*, and *Grade*, respectively. The new statements semantically describe the learning recommendation for the student (*?s*), including the video instruction related to both the Lab and the relevant learning resource (*?res*). The instance (*?g*) is marked by the data property *isRecommended*, meaning that the student's learning difficulty has been addressed, and a recommendation made.
- Lines 13–44 identify the learning context, indicating that the learner (*?s*), at the performance level *B*, cannot satisfy the question (*?q*) within the Lab (*?l*). The learning result is represented by the instance (*?g*), and its value is compared with the default value of the question (*?q*). Recommendations are only given to questions which have not been previously recommended. This status is reflected by the statement in line 27. The recommended learning materials include a related video instruction of the Lab (*?v*) and a learning resource with a difficult level (*?res*), which are randomly retrieved from the ontological base (lines 32–40). In order to serve the recommendation, an available *Recommendation* instance (*?rec*, lines 41-44) is selected for using in the rule consequently.

2.3 Building a Competitive and Interactive Learning Environment

SEMAG focuses on two important characteristics of the e-learning environment: competition and interaction. In competition, we propose a grading strategy which enables students to improve their learning performances by retaking the topic-quizzes several times throughout the duration of the course. In interaction, SEMAG provides two means:

1. instant generation of topic-quiz questions, based on student's performance and answers; and

2. personalized learning recommendations, which are generated through the implementation of the instructional semantic web rule-base.

2.3.1 Grading Strategy

The grading strategy is designed with the aim of encouraging students to participate positively in the learning process, in order to improve their performance. Student performance evaluations are divided to three levels: topic level, chapter level, and course level. In order to evaluate a student's understanding of the learning content unit (e.g., a topic, chapter, or course), we deliver a relevant quiz (e.g., topic quiz, chapter quiz, or final exam) following the unit. Student scores are calculated through Equation (1).

$$quizScore(u, quiz) = \frac{\sum_{i=1}^{n} score(u, q_i)}{n} \tag{1}$$

where n is the number of questions of in the *quiz*; and the $score(u, q_i)$ is the result which student u scored in question q_i of the *quiz*.

In the Topic level, the student u's score in a topic-quiz tq is computed through Equation (2).

$$topicScore(u, tq) = quizScore(u, tq). \tag{2}$$

In the Chapter level, the student u's total score in a chapter ch is computed through Equation (3).

$$chapterScore(u, ch) = \alpha \cdot \frac{\sum_{i=1}^{m} topicScore(u, tq_i)}{m} + (1 - \alpha) quizScore(u, chq) \tag{3}$$

where α is a predefined parameter ($0 \leq \alpha \leq 1$); m is the number of topic-quizzes in the chapter ch; tq_i is the i^{th} topic-quiz of the chapter ch; and chq is the chapter-quiz of chapter ch.

In the Course level, the student u's final score for the course c is computed through Equation (4).

$$finalScore(u, c) = \beta \cdot \frac{\sum_{i=1}^{k} chapterScore(u, ch_i)}{k} + (1 - \beta) quizScore(u, fe) \tag{4}$$

where β is a predefined parameter ($0 \leq \beta \leq 1$); k is the number of chapters in the course c; and fe is the course's final examination.

In order to build a more competitive learning environment, we allow students to retake the topic quizzes several times during the course, in order to improve their learning performances. Based on these settings, students are implicitly encouraged to study by themselves, encouraging active and positive participation.

2.3.2 Delivering Personalized Learning Recommendations

Personalized learning recommendation messages are built upon the *Recommendation* instances, written in the Vietnamese language, and were automatically based on message templates. These personalized recommendations are the reasoning results of instructional semantic web rules, and are sent to each student after the completion of each quiz. The social network and student's LMS mailbox are two channels for receiving personalized learning recommendation messages.

The algorithm, which uses the MAS to implement semantic reasoning and delivers personalized learning recommendations, is briefly described as follows. First, the Management agent asks the Student agent(s) to periodically update data from LMS database. Second, after receiving updated data from the Student agent(s), the Management agent asks the Teacher agent(s) to implement reasoning, by using the instructional semantic web rule-base. The Teacher agent(s) then send the reasoning results back to the Management agent, and the resulting *Recommendation* instances are pushed into the message queue. Lastly, while the message queue is not empty, the Management agent gets every *Recommendation* instance and transfers it to the Information agent(s). The Information agent is responsible for building recommendation messages, based on the templates, and delivering the messages to the recipients, via social network and LMS mailbox. The pseudo code of this process is shown in Algorithm 1.

Algorithm 1 Delivering personalized recommendation messages

```
procedure deliverMessages(student, teacher, information, queue)
    data ← updateData(student);                      ▷ student is a Student agent
    queue ← implementRules(teacher, data);           ▷ teacher is a Teacher agent
    while queue ≠ ∅ do                ▷ queue is a queue of Recommendation instances
        r ← pop(queue);                              ▷ r is a Recommendation instance
        deliver(information, r);                     ▷ information is an Information agent
    end while
end procedure
```

2.3.3 Generating Instant Topic-Quiz Questions

In this sub section, we present our knowledge-based solution, which uses the ontological base for generating personalized topic-quiz questions. Theoretically, a topic-quiz question is defined in Definition 2.

Definition 2 (Topic-quiz question)**.** Given q_i and t are the instances of the classes *Question* and *Topic* of the domain ontology O, respectively; where d is the level of difficulty of question q_i. A topic-quiz question tq, which relates to topic t and has a difficulty level of d, is defined as: $tq_{q_i,t,d} = \langle q_i, t, d\rangle$; in which, q_i satisfies the following condition: $relatesTo(q_i, t) \wedge hasDifficultLevel(q_i, d)$, whereas $relatesTo$ and $hasDifficultLevel \in R$, $R \in O$.

In general, the process of the proposed algorithm for generating instant topic-quiz questions is explained as follows: Given student s, whose learning performance of previous topic is p. In case u does not have a previous learning performance, the default value C will be assigned. Firstly, the algorithm generates an initial set of N topic-quiz questions, whereas each i^{th} difficult level has n_i questions (n_i are predefined parameters), based on the student's performance in previous topic. Secondly, these generated questions are presented one-by-one to student u, following an increasing level of difficulty, provided by the Information Agent. The student u's answers are collected by Student Agent. Given a topic-quiz question $tq_{q_i,t,d}$, if answered correctly, the next generated topic-quiz question will be presented to the student. If not, another topic-quiz question $tq_{q_j,t,d}$, ($j \neq i$) will be inferred by the Teacher Agent and introduced to student. The topic quiz will stop if one of the following ending conditions is satisfied:

1. student answers all of the generated questions correctly within the time scale which is defined in Equation (5) below; or
2. the time allotment has ended.

$$T = \sum_{i=1}^{k} time(q_i) \tag{5}$$

where k is the number of current generated topic-quiz questions and $time(q_i)$ is the default time of the question q_i. The pseudo-code of this algorithm is presented in Algorithm 2.

Algorithm 2 Generating instant topic-quiz questions

```
procedure GENERATEQUESTION(student, teacher, inform, u, p)
    QuestionSet ← generateInitialQuestions(u, p);        ▷ p is the performance of previous topic of student u
    for each q in QuestionSet do
        deliverQuestion(inform, q, u);                   ▷ inform is an Information agent
        answer ← getAnswer(student, q, u);               ▷ student is a Student agent
        if (answer is not correct) then
            tq ← inferSimilarQuestion(teacher, q, u);    ▷ teacher is a Teacher agent
            QuestionSet ← QuestionSet ∪ tq;
        end if
        if (Time is over) then
            exit;
        end if
    end for
end procedure
```

3 EXPERIMENTS

The predominant method of demonstrating the advantages of the ITS, is through a series of pretest-posttest experiments, which have been widely applied in recent

ITS studies [25, 29]. This method divides the participants into an experimental group and a control group. The former employs the ITS whereas the latter employs the non-ITS. The advantages of the ITS are proven through the comparison of the learning performances of the two groups. However, a weakness exists within this method, in that it cannot be compared with other ITS groups. In order to compare the proposed ITS with those of other related studies, relevant functional comparisons were applied [8]. Survey analyses were also implemented in order to evaluate the learners' comments regarding the efficiency of the proposed ITS [30]. Hence, for the purpose of demonstrating the advantages of SEMAG, we have adopted the pretest-posttest experiment model, the functional comparison method [8], as well as the survey-analysis method for evaluating ITS efficiency [30].

3.1 Experiment Settings

Our SEMAG system prototype was developed with the LMS Moodle, as it has been deployed for several years within our university, and includes many courses and valuable learning resources. And, given its vast popularity, Facebook was chosen for the purpose of spreading the learning space. The MAS construction was based on the Java-based agent platform JADE [2]. Jena[8], which is an open-source Java framework for building semantic-web linked data applications, was utilized to manage all ontological operations, as well as to implement the CONSTRUCT-based rules. Virtuoso[9], which is a multi-model data server supporting linked open data deployment, stored the knowledge base. The RestFB[10] library, which is a simple and flexible Facebook Graph API written in Java, was used to deliver personalized recommendations to the Facebook course group, while personalized messages were sent to students by inserting the message content into the relevant tables of the Moodle database. Figure 3 a) and 3 b) show examples of personalized learning recommendation messages which were sent to Facebook and Moodle, respectively.

All learning materials of this course were developed by the lecturer, and carefully selected from external sources. For instance, in order to guide students through lab deployment, the lecturer developed a series of video instructions[11], which were published in the easily accessed video-sharing website, YouTube. The instructional videos attracted over 1.5 million views at the time of this writing. Other web sources (e.g., SlideShare[12], Wikipedia[13], etc.) were selected by the lecturer to provide added learning materials to our learners. The learning materials were managed by the domain ontology, and several aspects of these rich informational sources were used to build many of the topic-based lessons.

---

[8] http://jena.apache.org/

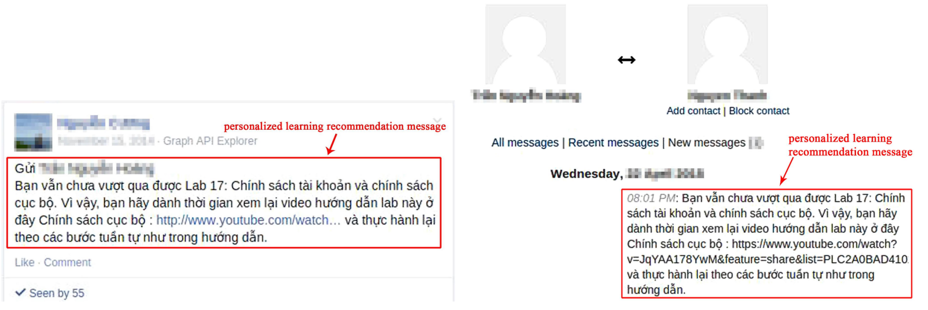
[9] http://virtuoso.openlinksw.com/

[10] http://restfb.com/

[11] http://www.youtube.com/ndhcuong

[12] http://www.slideshare.net/

[13] http://www.wikipedia.org/



a) Facebook message b) Moodle message

Figure 3. Examples of personalized learning recommendation messages

3.2 Evaluating Learning Performance

SEMAG was applied in teaching the subject of Networking offered to undergraduate students in the Business Information Systems major at Hue University, Viet Nam. The five-chapter course on the subject of Networking was taught over three months, from 15 September to 15 December, 2014. The participants consisted of 96 third-year students which were randomly assigned to either the control group or the experimental group; in which each group had 48 members. The control group utilized the non-ITS (Moodle), while the experimental group put into practice the proposed SEMAG. The two groups were taught by the same lecturer, and followed the same curriculum, course content, and lesson plans. The experimental design of the e-learning process is shown in Figure 4.

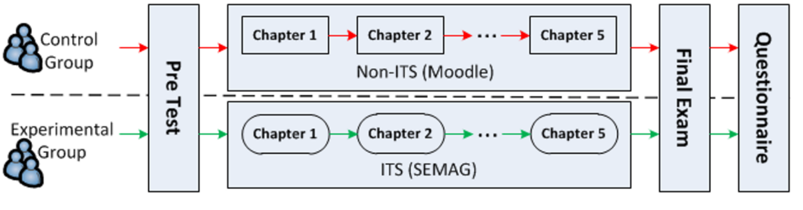


Figure 4. The experimental design

We calculated the t-test values (p-value) based on the participating students' scores. In addition, effect sizes were calculated using Cohen's d [7]. The pretest, designed with the target of checking the participants' general subject knowledge, was administered to all students, prior to the start of the course. At the conclusion of each topic, a topic quiz was delivered to the learners, and the learners' scores were collected. The students' learning achievements in each chapter's quiz were also recorded. Lastly, the results of the final exam were retrieved. All tests followed a similar 10-mark scale.

Group (size)	Pre-Test (m-score)	Chapter 1 (m-score)	Chapter 2 (m-score)	Chapter 3 (m-score)	Chapter 4 (m-score)	Chapter 5 (m-score)	Final Exam (m-score)
Ex-G (48)	3.02	5.64	7.13	7.42	8.75	8.65	8.06
Co-G (48)	2.84	5.53	6.53	6.88	8.06	7.77	7.13
t-(p-value)	0.385 (0.70)	0.511 (0.610)	2.206 (0.030)	2.509 (0.014)	2.908 (0.004)	3.169 (0.002)	2.518 (0.013)
Cohen's d	0.17	0.09	0.568	0.442	0.607	0.627	0.51

Table 1. Differences between the experimental and control groups in terms of mean scores, t-test results and Cohen's d. Ex-G: Experimental Group; Co-G: Control Group; m-score: mean score.

The t-test (p-value) of the pretest, chapter quizzes, and the final exam are presented in Table 1. We used the statistical software R [20] for data analysis. As shown in Table 1, the students of the two groups were assigned randomly, as there was no significant difference in the pretest results (p-value $= 0.70 > 0.05$ and Cohen's $d = 0.17 < 0.2$). The pretest results also indicated that the students had very little knowledge of networking prior to the course. After the first introductory chapter (Chapter 1), the learning performances of the two groups differed significantly in the remaining chapters (Chapters 2-5), as well as in the final exam. The results of the final exam in particular indicated a significant difference in learning performance between the two groups (p-value $= 0.013 < 0.05$ and Cohen's $d = 0.51 > 0.5$); the mean score of the experimental group peaked at 8.06, whereas that of the control group reached only 7.13. The students of the experimental group achieved higher learning performances than those of the control group; suggesting that the proposed SEMAG is capable of providing a better e-learning environment in general.

Group \ Day	1	2	3	4	5	6	7
Ex-G	48	25	17	19	15	9	8
Co-G	48	24	18	16	12	5	3

Table 2. Numbers of students retried the topic quiz of the topic: "4.4. Basic firewall settings with iptables". Ex-G: Experimental Group; Co-G: Control Group.

Specifically, the effectiveness of the personalized learning recommendations at topic-level generated by SEMAG may also be determined through the analysis of the learned topic quiz data, which includes students' learning results and their behavior logs. We randomly selected the topic quiz: "4.4. Basic firewall settings with iptables", and retrieved the results of both groups for comparison. In this example, the behavior logs of the topic quiz indicated that students of both groups had made several attempts to improve their learning performances, by repeatedly retaking the topic quiz, illustrated in Table 2. However, after seven days, there were no further attempts recorded from either group. The learning-performances of each group were compared, and are depicted in Figure 5. The experimental group recorded more A and B learning-performance levels than in the control group. Similarly, a greater number of lower learning-performance levels, including C, D and F, were achieved by the control group (non-ITS). The significant differences between the two groups

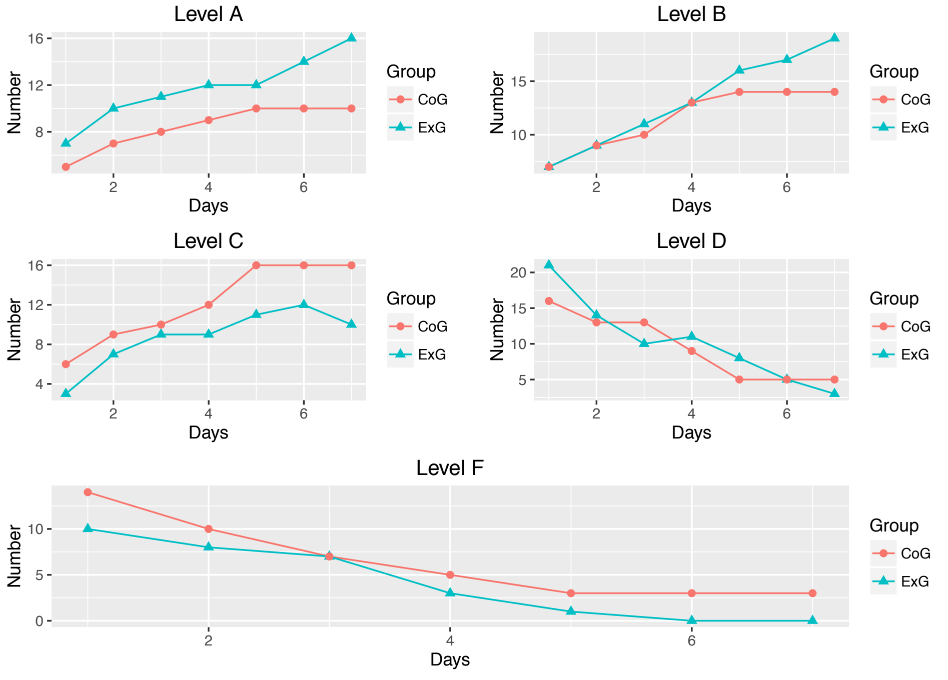


Figure 5. The topic-quiz results of the topic "4.4. Basic firewall settings with iptables". ExG: Experimental Group; CoG: Control Group.

were revealed primarily in the 5$^{\text{th}}$ to the 7$^{\text{th}}$ days. During this period, the students of the experimental group continued to improve their learning performances through independent effort, whereas few attempts were made by those in the control group, where the learning-performance lines of all levels remained stable. These results demonstrated the positive effects and overall success generated by the instant topic-quiz questions (produced by Algorithm 2) and the personalized learning recommendations (produced by Algorithm 1). In other words, the results confirm that SEMAG properly motivated students to participate independently and positively in the learning process.

3.3 Evaluating System Efficiency

In addition to comparing the learning results between the two groups, we also collected student comments regarding their usage of the e-learning systems, in order to assess and to deepen our understanding of their needs. Upon completion of the course, students were encouraged to give their comments by answering the question *"What are your comments about the e-learning system used in this course?"* via Google form. Within the seven day inquiry, we received 37 comments from the control group, and 42 comments of students in the experimental group. Comments were carefully analyzed, and categorized into six constructs; entitled Learning material, Quiz questions, Learning recommendation, Video instruction, Social network usage, and Moodle-based system usage, illustrated in Table 3.

Construct	Summarized Comments	No. of Responses (Ex-G)	No. of Responses (Co-G)
Learning material	The learning materials are useful and interesting.	34/36	29/32
Quiz question	Quiz questions are appropriate and help learners to recognize their weakness.	24/27	10/12
Learning recommendation	Learners receive useful and excited learning recommendations.	35/37	7/11
Video instruction	Video instructions help learners to implement network labs efficiently.	22/27	23/24
Social network usage	Receiving learning recommendation messages via Facebook makes closed contacts between learners and system, and encourages learners to use e-learning system.	37/39	0/0
Moodle-based system usage	Learners like to use Moodle-based system for learning and improving their learning results.	18/21	2/17

Table 3. Learners' positive comments about their e-learning systems

As shown in Table 3, the students of the experimental group offered more positive comments about Learning material, Quiz question and Learning recommendation than those of the Control group. These three constructs relate directly to the personalization of the learning recommendations; it therefore suggest that the students developed a greater appreciation for the SEMAG learning process. With a nearly perfect positive response rate (37/39), the preference of students in the experimental group to combine social networking in an e-learning environment, supports our anticipated results, and those of previous studies [4] and [21]. Examples of student feedback confirm their interest in receiving learning recommendation messages via Facebook: *"It is exciting because this is the first time I use Facebook for a learning purpose."*, *"I can keep contact with the e-learning system via the Facebook application on my smart phone without accessing the university's Moodle system via laptop."*, *"It is convenient to receive learning instructions via Facebook when I am surfing the Web."* and *"Facebook learning recommendation messages made me spend more time learning efficiently."* The most significant difference was revealed in the Moodle-based system usage. While many students of the experimental group claimed to enjoy using the university's Moodle-based system, only two individuals in the control group gave positive comments. Moreover, several negative comments were offered as well: *"It is bored to use the Moodle-based system."*, *"I think the learning style in Moodle is not suitable for me."* or *"I do not like to use Moodle system, but it is compulsory."* On the contrary, students of the Experimental group asserted their positive attitudes, for example: *"After following learning instructions via Facebook, I would like to access the Moodle system to improve my learning results, or to check my knowledge via topic quizzes."*, *"I think Moodle system became more active in this course."*, or *"I like the way Moodle generates question by question, in topic quizzes to test my understanding."* The student feedback supports the use of the SEMAG framework, which integrates the MAS to an available LMS, in order to build a new intelligent learning system. In other words, with SEMAG framework, students can enjoy a new ITS within an old LMS. The students of the Experimental group clearly demonstrated a heightened learning motivation in comparison to the students of the Control group (non-ITS). The added usage of a social network (Facebook) to widen the learning space further encouraged learners to learning efficiently. In summary, students were motivated to positively participate in the learning process through SEMAG.

4 RELATED WORK

In this section we review the state-of-the-art ITS studies, which can be categorized into two categories: the data-driven approach and the knowledge-based approach. We also investigate and discuss the significant differences between them and SEMAG.

Knowledge discovery in databases, or data mining, introduces various techniques in the process of turning data to knowledge [22], such as classification, clustering, and association rule mining. The data-mining approaches to ITS discovered knowledge in e-learning databases and built the knowledge models, which were used for the learning-recommendation process. For instance, different kinds of knowledge models were built by mining learning data such as clustering and association rule mining [23], association rules [16, 25] or decision trees, Naive Bayes and rules [10]. In case there are not enough learning data for training model(s), the data-mining approaches will reveal their weakness. In contrast to data-mining approaches, the knowledge model of SEMAG was built by transferring the lecturers' domain knowledge into machine-readable formats (ontology and rules). This method proved advantageous within the ITS domain, because the lecturer's knowledge and teaching experiences are critical resources, which help students to more efficiently improve their learning performance.

Within the knowledge-based approach, the use of ontology to formalize knowledge [9, 12] and the appearance of semantic web-based ITS studies [26, 27] have become popular in recent years. However, SEMAG differs in many respects. For instance, where Gescur [9] and SEMAG both specify domain knowledge by ontologies, and employ SPARQL queries to achieve their recommendation tasks, there are some significant differences between them:

1. SEMAG focuses on aiding learners, whereas Gescur's target users were teachers; and
2. the different target users led to different knowledge modeling techniques, different recommendation mechanisms, and different computing paradigms.

The most significant difference between SEMAG and the work of [12] involves student modeling. The authors of [12] designed the student model, based upon Bayesian networks and ontologies; whereas SEMAG creates a domain ontology to capture all of users' information. In addition, teachers' knowledge and experiences were not mentioned in the work of [12] but SEMAG did. For Protus 2.0 [26] and the work of Vidal-Castro et al. [27], the significant differences between SEMAG and these two ITS studies include:

1. Protus 2.0 [26] and the work of [27] expressed their rules by SWRL which is not as expressive as SPARQL-based rules of SEMAG; and
2. the learning space in SEMAG was spread to a social network, whereas the two aforementioned ITS studies limited learners, who confined in their web-based system.

ITS/ Approach of	Domain Ontology	Instructional Rules	Data Mining	MAS/ Agent Technology	LMS Integration	Social Network Integration
SEMAG	Yes	Yes	No	MAS	Yes	Yes
Romero et al. [23]	No	No	Yes	No	Yes	No
Lee et al. [16]	No	No	Yes	No	No	No
Protus system [25]	Yes	No	Yes	No	No	No
PDinamet system [10]	No	No	Yes	No	No	No
Gescur platform [9]	Yes	No	No	No	No	No
Grubisic et al. [12]	Yes	No	Yes	No	No	No
Protus 2.0 [26]	Yes	Yes	No	No	No	No
Vidal-Castro et al. [27]	Yes	Yes	No	No	No	No
Casamayor et al. [5]	No	No	Yes	Agent technology	Yes	No
INES [18]	Yes	No	No	Agent technology	No	No
X-Learn system [8]	Yes	No	No	MAS	No	No
Xu et al. [29]	No	No	No	MAS	No	No
Xu and Wang [30]	No	No	No	MAS	No	No
Yaghmaie and Bahreininejad [31]	Yes	No	No	Agent technology	Yes	No

Table 4. A summarized comparison between SEMAG and other related studies

Whether the chosen approach is data-mining or knowledge-based, the system infrastructure of the most recent ITS studies have been dominated by web-based systems, and/or multi-agent systems; because this architecture is easy to use, requires no-installation, and can be accessed web-wide. Furthermore, in order to produce personalized recommendations in a distributed and dynamic e-learning environment, several recent ITS studies [5, 8, 18, 29, 30, 31] used agent technology to reduce server workload. These agents, containing properties like autonomy, preactivity, pro-activity, and social ability [28], are responsible for generating learning recommendations, and can therefore help to reduce the teacher's workload, as well as increase learner-system interaction. SEMAG and these agent-based ITS studies share this common target; however, the most significant difference between them is rooted in the recommendation-making process. SEMAG focuses on improving learners' performances in higher levels (chapter or course levels), much like the agent-based ITS studies; however, also helps learners to improve their performances at lower levels (topic levels). Furthermore, the knowledge-based generation of instant topic-quiz questions makes SEMAG a dynamic, interactive, and competitive learning environment. A summarized comparison between SEMAG and the aforementioned research is presented in Table 4.

5 CONCLUSION

In this paper, we present SEMAG, a novel semantic-agent learning recommendation mechanism. The core of SEMAG's novel recommendation-making process consists of two reasoning algorithms, which are responsible for building a competitive and interactive learning environment, and provides personalized learning recommendations and questions, instantly and dynamically. A social network is also integrated into SEMAG architecture, in order to increase the students-ITS interaction. Experiments produced promising results, which supported the favorable use of SEMAG. In future areas of study and refinement of the SEMAG architecture, we plan to address and target more adaptively personalized learning recommendations.

REFERENCES

[1] Arevalillo-Herráez, M.—Arnau, D.—Marco-Giménez, L.: Domain-Specific Knowledge Representation and Inference Engine for an Intelligent Tutoring System. Knowledge-Based Systems, Vol. 49, 2013, pp. 97–105, doi: 10.1016/j.knosys.2013.04.017.

[2] Bellifemine, F.—Poggi, A.—Rimassa G.: JADE – A FIPA-Compliant Agent Framework. Proceedings of the Practical Applications of Intelligent Agents, London, 1999, pp. 97–108.

[3] Bittencourt, I.—Costa, E.—Silva, M.—Soares, E.: A Computational Model for Developing Semantic Web-Based Educational Systems. Knowledge-Based Systems, Vol. 22, 2009, No. 4, pp. 302–315, doi: 10.1016/j.knosys.2009.02.012.

[4] Bosch, T.: Using Online Social Networking for Teaching and Learning: Facebook Use at the University of Cape Town. Communicatio: South African Journal for Communication Theory and Research, Vol. 35, 2009, No. 2, pp. 185–200.

[5] Casamayor, A.—Amandi, A.—Campo, M.: Intelligent Assistance for Teachers in Collaborative E-Learning Environments. Computers and Education, Vol. 53, 2009, No. 4, pp. 1147–1154, doi: 10.1016/j.compedu.2009.05.025.

[6] Chi, Y. L.: Ontology-Based Curriculum Content Sequencing System with Semantic Rules. Expert Systems with Applications, Vol. 36, 2009, No. 4, pp. 7838–7847.

[7] Cohen J.: Statistical Power Analysis for the Behavioral Sciences. 2nd edition. Lawrence Erlbaum Associates, 1988.

[8] De Meo, P.—Garro, A.—Terracina, G.—Ursino, D.: Personalizing Learning Programs with X-Learn, an XML-Based, "User-Device" Adaptive Multi-Agent System. Information Sciences, Vol. 177, 2007, No. 8, pp. 1729–1770.

[9] Fernández-Breis, J. T.—Castellanos-Nieves, D.—Hernández-Franco, J.—Soler-Segovia, C.—del Carmen Robles-Redondo, M.—González-Martínez, R.—Paz Prendes-Espinosa, M.: A Semantic Platform for the Management of the Educative Curriculum. Expert Systems with Applications, Vol. 39, 2012, No. 5, pp. 6011–6019, doi: 10.1016/j.eswa.2011.11.123.

[10] Gaudioso, E.—Montero, M.—Hernandez-del-Olmo, F.: Supporting Teachers in Adaptive Educational Systems Through Predictive Models: A Proof of Concept. Expert Systems with Applications, Vol. 39, 2012, No. 1, pp. 621–625, doi: 10.1016/j.eswa.2011.07.052.

[11] Gladun, A.—Rogushina, J.—García-Sanchez, F.—Martínez-Béjar, R.—Fernández-Breis, J. T.: An Application of Intelligent Techniques and Semantic Web Technologies in E-Learning Environments. Expert Systems with Applications, Vol. 36, 2009, No. 2, Part 1, pp. 1922–1931.

[12] Grubišić, A.—Stankov, S.—Peraić, I.: Ontology Based Approach to Bayesian Student Model Design. Expert Systems with Applications, Vol. 40, 2013, No. 13, pp. 5363–5371, doi: 10.1016/j.eswa.2013.03.041.

[13] Horrocks, I.—Patel-Schneider, P. F.—Boley, H.—Tabet, S.—Grosof, B.—Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission, 2004, pp. 21–79.

[14] Kirschner, P. A.—Karpinski, A. C.: Facebook® and Academic Performance. Computers in Human Behavior, Vol. 26, 2010, No. 6, pp. 1237–1245, doi: 10.1016/j.chb.2010.03.024.

[15] Lasierra, N.—Alesanco, A.—O'Sullivan, D.—García, J.: An Autonomic Ontology-Based Approach to Manage Information in Home-Based Scenarios: From Theory to Practice. Data and Knowledge Engineering, Vol. 87, 2013, pp. 185–205, doi: 10.1016/j.datak.2013.06.004.

[16] Lee, C.-H.—Lee, G.-G.—Leu, Y.: Application of Automatically Constructed Concept Map of Learning to Conceptual Diagnosis of E-Learning. Expert Systems with Applications, Vol. 36, 2009, No. 2, Part 1, pp. 1675–1684.

[17] Maramis, C.—Falelakis, M.—Lekka, I.—Diou, C.—Mitkas, P.—Delopoulos, A.: Applying Semantic Technologies in Cervical Cancer Research. Data and Knowledge Engineering, Vol. 86, 2013, pp. 160–178, doi: 10.1016/j.datak.2013.02.003.

[18] Mikic Fonte, F. A.—Burguillo, J. C.—Llamas Nistal, M.: An Intelligent Tutoring Module Controlled by BDI Agents for an E-Learning Platform. Expert Systems with Applications, Vol. 39, 2012, No. 8, pp. 7546–7554, doi: 10.1016/j.eswa.2012.01.161.

[19] Noy, N.—McGuinness, D. L.: Ontology Development 101. Knowledge Systems Laboratory, Stanford University, 2001.

[20] R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[21] Roblyer, M. D.—McDaniel, M.—Webb, M.—Herman, J.—Witty, J. V.: Findings on Facebook in Higher Education: A Comparison of College Faculty and Student Uses and Perceptions of Social Networking Sites. The Internet and Higher Education, Vol. 13, 2010, No. 3, pp. 134–140, doi: 10.1016/j.iheduc.2010.03.002.

[22] Romero, C.—Ventura, S.: Educational Data Mining: A Survey from 1995 to 2005. Expert Systems with Applications, Vol. 33, 2007, No. 1, pp. 135–146.

[23] Romero, C.—Ventura, S.—Zafra, A.—de Bra, P.: Applying Web Usage Mining for Personalizing Hyperlinks in Web-Based Adaptive Educational Systems. Computers and Education, Vol. 53, 2009, No. 3, pp. 828–840.

[24] Rowe, M.—Fernandez, M.—Angeletou, S.—Alani, H.: Community Analysis Through Semantic Rules and Role Composition Derivation. Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 18, 2013, No. 1, pp. 31–47.

[25] Vesin, B.—Klašnja-Milićević, A.—Ivanović, M.—Budimac, Z.: Applying Recommender Systems and Adaptive Hypermedia for E-Learning Personalization. Computing and Informatics, Vol. 32, 2013, No. 3, pp. 629–659.

[26] Vesin, B.—Ivanović, M.—Klašnja-Milićević, A.—Budimac, Z.: Protus 2.0: Ontology-Based Semantic Recommendation in Programming Tutoring System. Expert Systems with Applications, Vol. 39, 2012, No. 15, pp. 12229–12246.

[27] Vidal-Castro, C.—Sicilia, M.-Á.—Prieto, M.: Representing Instructional Design Methods Using Ontologies and Rules. Knowledge-Based Systems, Vol. 33, 2012, pp. 180–194, doi: 10.1016/j.knosys.2012.04.005.

[28] Wooldridge, M.: An Introduction to MultiAgent Systems. 1st edition. John Wiley and Sons, New York, June 2002.

[29] Xu, D.—Huang, W. W.—Wang, H.—Heales, J.: Enhancing E-Learning Effectiveness Using an Intelligent Agent-Supported Personalized Virtual Learning Environment: An Empirical Investigation. Information and Management, Vol. 51, 2014, No. 4, pp. 430–440.

[30] Xu, D.—Wang, H.: Intelligent Agent Supported Personalization for Virtual Learning Environments. Decision Support Systems, Vol. 42, 2006, No. 2, pp. 825–843.

[31] Yaghmaie, M.—Bahreininejad, A.: A Context-Aware Adaptive Learning System Using Agents. Expert Systems with Applications, Vol. 38, 2011, No. 4, pp. 3280–3286, doi: 10.1016/j.eswa.2010.08.113.

Cuong Dinh Hoa Nguyen received his Ph.D. degree of computer science in 2016 from Khon Kaen University, Thailand. His research interests include e-learning, intelligent tutoring systems, semantic web technologies, data mining and recommender systems. Now he is Lecturer at Department of Business Information System, College of Economics, Hue University, Viet Nam.

Ngamnij Arch-int received her Ph.D. degree in computer science from Chulalongkorn University, Thailand in 2003. She is currently Assistant Professor at the Department of Computer Science at Khon Kaen University, Thailand. Her research interests include the semantic web, web services, semantic web services, and heterogeneous information integration.

Somjit Arch-int received his Ph.D. degree in computer science from the Asian Institute of Technology, Thailand in 2002. He is currently Associate Professor at the Department of Computer Science, Khon Kaen University, Thailand. His previous experiences include the development of several industry systems and consulting activities. His research interests are business component-based software development, object-oriented metrics, ERP/Logistics and supply chain, traceability system and semantic web. He is a member of IEEE Computer Society.