Evaluation of an Automatic Question Generation Approach Using Ontologies

Noor Hasimah Ibrahim Teo and Mike Joy
Department of Computer Science, University of Warwick, UK
N.H.Ibrahim-teo@warwick.ac.uk
M.S.Joy@warwick.ac.uk

Abstracts: Advancements in Semantic Web techniques have led to the emergence of ontology based question generation. Ontologies are used to represent domain knowledge in the form of concepts, instances and their relationships as their elements. Many research strategies for generating questions using ontologies have been proposed but little work has been done on investigating whether an ontology is an appropriate source of data for question generation. Since there is no standard guideline for developing an ontology, the representation of ontology elements might vary in many ways, and this paper aims to investigate how the naming of ontology elements would affect the questions generated. In order to achieve this aim, two research questions will be investigated which are: how many correct questions can be generated from an ontology, and what are the reasons for incorrect questions being generated. Categorized question templates and a set of question strategies for mapping templates with a concept in an ontology are proposed. A prototype has been developed with a *Reader* to read data from input file and 3 question generators namely *termQG, ClassQG and PropertyQG* to generate questions for 3 ontological approaches. After questions have been generated, the number of correct questions generated is calculated and the reasons for incorrect questions are identified. Two ontologies have been used, an Operating System Ontology and a Travel Ontology. Twenty question templates from three question categories – *definition, concept completion* and *comparison* – together with 5 question generation strategies have been used in this evaluation. Results shows that more than half of the questions generated are correct and there are 3 distinct reasons why incorrect questions may be generated. The main contribution to incorrect question generation was inappropriate naming of ontology elements where 4 distinct categories are further identified. In addition, evaluation shows that the object type should be considered when designing question templates. Furthermore, the evaluation indirectly shows the effectiveness of the ontological approaches for generating questions from a real-world ontology.

Keywords: question generation, ontology, ontological approaches, assessment, question template

1. Introduction

Recently, machine generated questions research have been widely studied. Developing automatic methods for Question Generation (QG) can alleviate the burden of both traditional and e-Learning assessments. Unstructured text and ontologies are two sources that are typically used for QG.

Advancements in Semantic Web technologies have led to the emergence of ontology based question generation. Reasons for question generation include formative and summative assessment (Papasalouros et al., 2008, Mitkov et al., 2006), exercise questions (Lin and Chou, 2011), problem solving questions (Kumar, 2005), domain specific general questions (Soldatova and Mizoguchi, 2009), or general questions (Hovy et al., 2001). Different strategies have been used to transform semantic information from an ontology into questions. The system proposed by Afzal and Mitkov (2014) uses three strategies to generate Multiple Choice Questions (MCQs) from a domain ontology which are class based strategies, property based strategies, and terminology based strategies. OntoQue (Al-Yahya, 2011, Al-Yahya, 2014) claims to be using a semantic based approach where it uses knowledge in the ontology about domain entities, such as classes for MCQs and true/false, properties for fill-in the blank (FIB) questions, to generate semantically correct assessment items. Papasalouros et al. (2008) have introduced three approaches for ontology-based single response MCQ generation which are class based, property-based and terminology-based.

Most of the works on ontology QG have applied similar ontological approaches for generating questions (class-based, property-based and terminology-based) and look mainly into how to generate distractors for MCQs and similar types of questions. Therefore, in our research we aim to explore factual question generation for short/long answer types of question using the same approaches. As a first step to achieving this aim, we investigate whether an ontology is an appropriate source for generating questions automatically. For this purpose, we will investigate the following research questions:

- How many correct questions can be generated from an ontology?
- What are the reasons for incorrect questions being generated?

The discussion in this paper will begin with a discussion of related works on QG in section 2. Section 3 discusses the method undertaken to conduct this experiment and the last part discusses the results obtained from the evaluation.

2. Question generation (QG)

2.1 What is a correct question?

A question is a sentence that uses to make a request of information. In English, there are three types of sentences: simple, compound and complex. A simple sentence usually has basic elements, including a subject, a verb and a clear meaning. A question Q is a type of sentence that is made up from 3-tuples which are question word (QW), predicate (P), and key element (KE).

A Question Word is a keyword used to indicate the type of information needed, a Predicate is the verb to express the action of the subject, and subject is normally a noun in a sentence. For example:

1. "What is an operating system?"
   QW: What
   *P**:** is*
   KE: Operating System

2. Explain the advantage of Shortest Job First algorithm."
   QW: Explain
   P: the advantage of
   KE: Shortest Job First algorithm

3. In the context of memory management, what is address binding?
   QW: In the context of…. what
   P: is
   KE: memory management, address binding.

A correct question can be defined as question that has no ambiguity and the question word is suitable to be used with the concept for generating the question.

2.2 Question taxonomies

Graesser and Person (1994) have divided questions for a tutorial into four categories: questions with *short answers,* questions with *long answers, assertions*, and *requests or directive questions*. Short answers are further classified into several subcategories which are: *verification, disjunctive, concept completion, feature specification* and *quantification*, whereas long *answers* are classified into *definition, example, comparison, interpretation, causal antecedent, causal consequence, goal orientation, instrumental/procedural, enablement*, and *expectation judgemental*. The authors claim the last 5 question types from the list are highly correlated with the deeper level of cognition in Bloom's taxonomy. The authors also suggest three question types- *pumps*, *hints*, and *prompt*s - that bind with low, medium and high levels of specificity (Graesser, 1995). Olney, Grasser and Person (2012) proposed three questioning strategies which are *contextual verification*, *forced choices*, and *causal chain*. Both contextual and forced choice questions use features extracted from two related concepts for generating question stems.

Bodoff and Raban (2015) adapt the question classification by Graesser and Person for investigating relationships between question types, elicitations and fees that are paid for mediated search services. They consider only four categories of the Graesser and Person (1994) taxonomy due to the kind of questions encountered in their data sample. Apart from the four categories – concept completion, enablement, instrumental/procedural, and quantification – a new *identity* question type was introduced which was found to be reliable in predicting elicitation and fees.

The web-based collaborative argumentation system proposed by Le et al. (2014) is another example of a recent question generation application that uses Graesser and Person's question classification. This work proposed a technical approach to assist students in understanding a given discussion topic by means of generating questions for argumentation with the use of semantic information. This work employed syntax-based, template-based and semantic based approaches where WordNet and Wikipedia are used to provide the semantic information to generate semantic question. Question templates are defined according to Heilman (2011) and are constructed for each of 9 categories as defined by Graesser and Person. Each placeholder in the question template is replaced with a noun or noun phrase from a discussion topic. The question categories enable the overall topic to be covered for argumentation.

2.3 Ontologies for QG

The notion of ontology, present within Artificial Intelligence (AI) research, focuses on the conceptualization or categorization process, prior to the actual building of a knowledge base. The continuous growth of the web that has integrated with business and personal life has caused ontology research to focus on the web. The Paper on 'Ontology Development 101' (Noy and McGuinness, 2001) mentions several reasons why ontologies are needed in the web. According to Noy and McGuinness, an ontology is a formal specification of conceptualization (Gruber, 1993) that can be used to share and enable reuse of domain knowledge.

Recent research has led to the emergence of ontology based QG. The reasons for QG include formative and summative assessment (Papasalouros et al., 2008, Mitkov et al., 2006), repetitive exercises (Lin and Chou, 2011), problem solving (Kumar, 2005), domain specific questions (Soldatova and Mizoguchi, 2009), and general questions (Hovy et al., 2001).

Substantial research efforts are being made in the generation of ontology based MCQ item generation. They define new strategies to identify key answers and distractors for MCQs. Text similarity measures and ontology elements are measured to identify keys and distractors (Cubric and Tosic, 2011), and an item that has the highest similarity will become the key answer and an item with a lower similarity index is chosen as a distractor for the MCQ. Key answers can also be selected from an RDF object and the distractors selected randomly from a collection of items under the same class as the key (Al-Yahya, 2014).

The proposed approach generates questions according to a specified ontology represented in OWL and hence can be used to generate questions for different ontologies as compared to a template based approach that might restrict the choice of question to be asked. Results from experiments have showed that class-based and terminology based approaches provide better syntactically correct questions but produce fewer questions compared to property based questions. The work focuses on creating the correct (key answer) and incorrect statements (distractors) for MCQs question that only asked 'Choose the correct sentence'.

Another engine that is capable of generating MCQs using a domain ontology is OntoQue (Al Yahya, 2011). Besides MCQs, the system is capable of generating true/false and FIB styles of questions. OntoQue has said to be using a semantic based approach where it uses knowledge from the ontology about domain entities such as classes for MCQs and true/false, and properties for FIB questions to generate semantically correct assessment items. Most questions generated are reported to be classified as good questions based on high precision data obtained experimentally. However the majority of questions constructed using this system are at the first level of Bloom taxonomy. MCQ stems are formulated by asking the instance of the class only and the key taken from the object of the statement.

3. Method

3.1 Generate question templates

In order to generate question templates, first a total 0f 259 questions from an Operating System textbook (Silberschatz, Galvin and Gagne, 2012) were collected and analysed. For this evaluation, we only selected factual question of three categories from the Graesser and Person Taxonomy which were *definition, concept completion* and *comparison*. After classifying the question, we proceeded with formalizing the questions using predicate logic to obtain question patterns. This process is called *semantic interpretation of questions into formal representation*. During this process, each key element from a sentence is extracted and replaced with a placeholder to create question stems. After that, we remove any duplicate pattern of question stems and identify a list of categorized unique templates as in Table 1.

Table 1: Categorized question templates

Category	Template
Definition	Define [X]. What is [X]? What does it means by [X]? What is a [X] in [SC]? Define a [X] in [SC]? In the context of [SC], what is [X]? What is meant by the term [X] in a [SC]?
Concept Completion	Explain the term [X]. Discuss what is [X]. Within the context of [SC], explain the term [X]. What [P] [X]? Explain [P] [X]. In the context of [SC], what [P] [X]?
Comparison	Differentiate between [X] and [Y]. What is the difference between [X] and [Y]? How does [X] differ from [Y]? Describe the differences between [X] and [Y]. Compare between [X] and [Y]. With regards to [SC], explain the difference between [X] and [Y].

Notes: [X] and [Y] is a concept. [SC] is a superclass of [X] or [Y]

3.2 Generate question strategies

This process was conducted before we removed the duplication in the question templates. We took the list of question stems and identified how each key element in the questions is represented in the ontology and how one key element relates with another by checking the relationship that exists between relevant matching concepts in the ontology. Next, we used three ontological approaches proposed by Papasoulourus et al. (2008) in this evaluation and categorized our question template accordingly into these approaches. In order to fit the questions into the approaches, we proposed a set of strategies in which each of the strategy is defined as follows:

<u>Term Based Approach</u>
Strategy 1: Concept only

- The question can be generated if there is a concept C which exists in an ontology.

<u>Class Based Approach</u>
Strategy 2: Is-a relation

- The question can be generated if a concept C1 is a subclass of another concept C2 which exist in an ontology.

Strategy 3: Cross relation

- The question can be generated if there is a concept C1 that is subclass of concept C and concept C2 is also a subclass of concept C which exists in an ontology.

<u>Property Based Approach</u>
Strategy 4: Term + Property

- The question can be generated if there is a concept C that has object property P which exists in an ontology.

Strategy 5: Class + Property

- The question can be generated if there is a concept C1 that is related to concept C2 by object property P and which exist in an ontology.

Table 2 shows the results of further classification of question templates into the three ontological approaches. As shown in the table the first approach is for questions with only one key element. The second approach is for questions with more than one key element but for this evaluation we only consider two key elements to reduce the complexity of identifying relations in the ontology. The third approach is for questions with predicates other than 'is' or 'are' as these predicate we assume are already represented as hierarchical relations in the ontology.

Table 2: Further classification of question templates

Approach	Category	Template
Term-based	Definition	Define [X]. What is [X]? What does it means by [X]?
	Concept Completion	Explain the term [X]. Discuss what is [X].
	Comparison	*[no template]*
Class-based	Definition	What is a [X] in [SC]? Define a [X] in [SC]? In the context of [SC], what is [X]? What is meant by the term [X] in a [SC]?
	Concept Completion	Within the context of [SC], explain the term [X] .
	Comparison	Differentiate between [X] and [Y]. What is the difference between [X] and [Y]? How does [X] differ from [Y]? Describe the differences between [X] and [Y]. Compare between [X] and [Y]. With regards to [SC], explain the difference between [X] and [Y].
Property-based	Definition	*[no template]*
	Concept Completion	What [P] [X]? Explain [P] [X]. In the context of [SC] , what [P] [X]?
	Comparison	*[no template]*

Notes: [X] and [Y] is a concept. [SC] is a superclass of [X] or [Y]

3.3 Preparing data input and design knowledge base

There are two data inputs used for this experiment which are ontologies and further categorised question templates. Both data inputs are stored in the form of text files. For ontologies, triples are extracted from an ontology in the form of "Concept#1 is-a Concept#2", while the question templates are written in the form as in Table 2 for the ontology's concept instantiation later.

3.4 Design and code algorithm

There are four components in this system: *Reader*, *TermQuestionGenerator ClassQuestionGenerator* and *PropertyQuestionGenerator*. Figure 1 shows the system architecture which includes three components for question generation in the Question Generator and one for reading data from the data file.

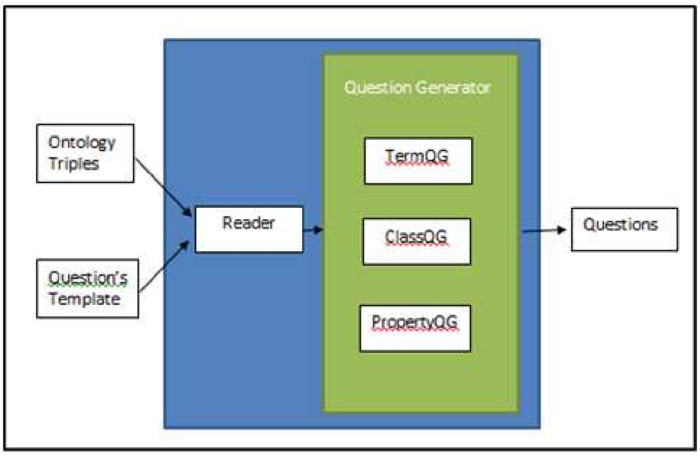

Figure 1: System architecture

3.5 Generate question for different ontology

There were 2 ontologies used for this evaluation: Operating System Ontology (OSO) (Silberschatz, Galvin, and Gagne, 2012) and Travel Ontology (TO) (Knublauch, 2004). The OSO contains 97 concepts and TO contains 41 concepts. OSO is the top level ontology with only hierarchical types of relation and TO is a descriptive type of ontology with hierarchical type data structure and object properties.

3.6 Evaluation

The evaluation was conducted and the results reported using simple statistical calculations. First, the generated questions were analysed to determine correct and incorrect questions. At the same time each incorrect question was provided with a reason. Finally, all the incorrect questions were categorized based on the pattern of the identified reasons.

The next section will discuss the results obtained from this evaluation in twofold, which are:

- The numbers of correct questions generated;
- The reasons why incorrect questions were generated (if any).

4. Results

4.1 How many correct questions can be generated from an ontology?

Table 1 shows the numbers of total questions generated from both ontologies with OSO generating three times higher numbers of question compared to TO even though the number of concepts in OSO is only double those in TO.

Table 3: Numbers of question generated

Ontology	Ontology elements	Number of questions generated by category			Total of Question Generated
		Definition	Concept Completion	Comparison	
OSO	C: 97 HR: 86 OPR:NA	Term: 97 Class: 86 Property: NA	Term: 97 Class: 86 Property: NA	Term: NA Class: 198 Property: NA	564
TO	C: 41 HR: 26 OPR:31	Term: 41 Class: 26 Property: NA	Term: 41 Class: 26 Property: 31	Term: NA Class: 17 Property: NA	182

Figure 2 shows the percentages of correct questions that can be generated from ontologies where OSO gives more promising results compared to TO. OSO shows a higher percentage of correct questions in the first two question categories, namely definition and concept completion, but a slightly lower percentage compare to TO in the third question category, comparison. The result from this experiment has shown that more than three quarters of incorrect questions generated from both ontologies are due to incorrect naming of concepts. We list all the reasons of incorrect naming and organize them into five categories in the following subsection.

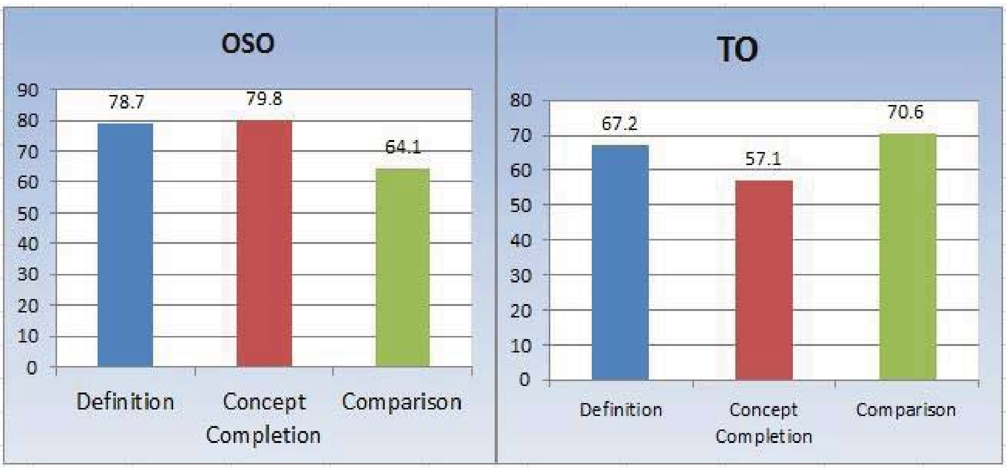

Figure 2: Percentages of correct question generated

4.2 What are the reasons for incorrect questions generated?

The result from this evaluation shows 3 main reasons for incorrect question generation which are:

- Inappropriate representation of a concept's name;
- Inappropriate use of question templates;
- Inappropriate representation of object properties.

Details of each reason are discussed in following subsection.

4.2.1 Inappropriate representation of a concept's name

Four different subcategories are identified, and the following discusses each of the categories with examples, why it causes problems and possible solutions.

4.2.2 Name of the concept is too long.

Example:

interaction-between-processes-and-o-s in process-and-thread-management

Why it causes problems:

This kind of naming will give too much information in one concept. It will be difficult to control and to reason about. It creates problem to generate questions as this representation include nouns as well as action verbs in one word. For example, if we generate question using strategy 1, it may generate for example:
"Define *interaction-between-processes-and-o-s in process-and-thread-management".*

Possible solution:

Use simple names for concepts.

4.2.3 A concept name contains two or more nouns.

Example:

O-s-services-and-components

Why it causes problems:

This kind of naming will introduce ambiguity as we are not sure of which concept is the key element for the question. For example, if we generate a question using strategy 1, it may generate:

"*Define o-s-services-and-components*."

Possible solution:

Use single concepts by splitting the compound concept into two.

4.2.4 Repeating the word used for sub and super concept.

Example:

Security *problem is-a* ***security***

Why it causes problems:

Although it is not wrong to represent concepts in this way, but by applying the proposed strategies it will generate redundant words in the question. For example:

"Explain security problems in security"

It would be better rephrased as:

"Explain problems in security"

Possible solution:

As the superclass already mentioned is about 'security', it is understood if the subclass only uses the word 'problem'. Alternatively, we can eliminate keywords that already exist in the superclass in the code.

4.2.5 Naming of concept includes action verb.

Example:

Implementation-of-access-matrix

Why it causes problems:

This kind of word is not suitable for naming concepts. Normally a concept will use a noun or noun phrase for representation. When we try to instantiate a definition type of question template that has pattern "Define [X]." it will give:

"*Define implementation-of-access-matrix."*

However, if we represent the concept using noun (access-matrix) and action verb (implementation-of) as object properties then we get questions like "*Define access-matrix*." by using Strategy 1 and "*Explain the implementation-of access-matrix*." by using Strategy 4.

Possible solution:

The action verb is more suitable to be used as an object property instead of combining with a concept word.

4.2.6 Inappropriate use of question templates

In contrast, the incorrect questions generated from TO are not limited to inappropriate ontology representations but also due to other aspects. For example, in TO there is a concept name 'Sydney' which is unsuitable to use a definition type of question as people normally do not ask the definition of a location. The most appropriate question word for location would be 'where', and this will change the category of question to concept completion. From this result, we can conclude that the question word must depend on object type and thus each question word might only be suitable for certain question types. For example, a question like 'Define Sydney' or 'Define Hotel' uses an incorrect question word, and in this case the question word should be replaced with 'Where is'.

4.2.7 Inappropriate representation of object properties

In addition, inappropriate word use for an object property will affect the meaning of the question that will be generated. For example, the triple 'beach has-a Sydney' in TO sounds semantically incorrect and a question generated like "What beach has-a Sydney?" from this triple will automatically be incorrect. It will change the meaning of the sentence. It will be appropriate if the object property has been changed to *is-in* instead of *has-a*. Therefore, appropriate names of object properties are as important as concept names to provide semantically correct representations.

5. Discussion

Overall, the investigation shows promising and satisfactory results when used with OSO. Most of the incorrect questions were due to inappropriate names to represent the concepts. This problem is reduced if the ontology representation uses a proper naming convention. On the other hand, the evaluation shows satisfactory results when used with TO. This is due to the question template having unsuitable object types for the concepts in TO. There are four different object types which bind correctly with suitable question words: *entity (what), location (where), people (who)* and *time (when).* OSO mainly represents concepts using entity object types, while TO has a combination of entity and location. The question templates used in this evaluation work well with the entity type of object.

In addition, the OBO Foundry Ontology (OBO Foundry, 2009), has proposed a set of naming conventions to help ontology developers and users to avoid flaws and inaccuracies when editing, matching and also help to reduce readability problems. The results from this experiment have shown that inappropriate naming of concepts in ontology also results in incorrect question generated. The suggestion is to use explicit and concise names which are kept short and memorable, but precise enough to capture the intended meaning. The result from the experiment shows this to be the most frequently occurring problem resulting in incorrect questions generated.

As a conclusion, we conclude that the experiment result has shown that the approach as well the strategies proposed are able to generate correct questions provided the ontology elements use proper and concise names or follow naming conventions suggested by the OBO Foundry. We also found that the object types should be identified during question template generation to indicate what question words are suitable for the concepts. The results can be improved by finding the solution for the errors found in this evaluation.

Acknowledgements

This research was supported by Universiti Teknologi MARA and Ministry of Education, Malaysia.

References

Afzal, N. and Mitkov, R. (2014). Automatic Generation of Multiple Choice Questions Using Dependency-Based Semantic Relations. *Soft Computing,* 18**,** 1269-1281.

Al-Yahya, M.(2011) Ontoque: A Question Generation Engine for Educational Assesment Based on Domain Ontologies. *11th IEEE International Conference on Advanced Learning Technologies (ICALT*). IEEE, 393-395.

Al-Yahya, M. (2014). Ontology-Based Multiple Choice Question Generation. *The Scientific World Journal,* 2014**,** 9 pages.

Bodoff, D. and Raban, D. (2015). Question Types and Intermediary Elicitations. *Journal of the Association for Information Science and Technology*.

Cubric, M. and Tosic, M. (2011). Towards Automatic Generation of e-Assessment using Semantic Web Technologies. *International Journal of e-Assessment*, 1, 1.

Graesser, A.C. and Person, N.K. (1994). Question Asking During Tutoring. *American Educational Research Journal,* 31**,** 104-137.

Graesser, A.C., Person, N.K. and Magliono, J.P. (1995). Collaborative Dialogue Patternds in Naturalistic One-on-One Tutoring. *Applied Cognitive Psychology*, 9, 495-522.

Heilman, M. (2011) Automatic Factual Question Generation from text. PhD thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University.

Hovy, E., Gerber, L., Hermjakob, U., Lin, C.-Y. and Ravichandran, D. (2001) Toward Semantics-Based Answer Pinpointing. *Proceedings of the First International Conference on Human Language Technology Research, 2001*. Association for Computational Linguistics, 1-7.

Knublauch, H. (2004). Ontology-driven software development in the context of the semantic web: An example scenario with Protege/OWL. In *1st International Workshop on the Model-Driven Semantic Web (MDSW2004)* (pp. 381-401). Monterey, California, USA.

Kumar, A. N. (2005). Generation of Problems, Answers, Grade, and Feedback Case Study of a Fully Automated Tutor. *Journal on Educational Resources in Computing (JERIC),* 5**,** 3.

Le, N.-T., Nguyen, N.-P., Seta, K. and Pinkwart, N. (2014). Automatic Question Generation for Supporting Argumentation. *Vietnam Journal of Computer Science,* 1**,** 117-127.

Lin, C.-Y. & Chou, C.-C.(2011) Effectiveness of Ontology-Based Online Exam Platform for Programming Language Education. *Computing in Civil Engineering*, 2011. ASCE, 907-914.

Mitkov, R., An Ha, L. and Karamanis, N. (2006) A Computer-Aided Environment for Generating Multiple-Choice Test Items. *Natural Language Engineering,* 12**,** 177-194.

Noy, N.F., McGuinness, D.L. (2001) Ontology Development 101: A guide to creating your first ontology. Technical Report SMI-201-0880, Stanford Medical Informatics.

Olney, A. M., Graesser, A. C. and Person, N.K. (2012). Question Generation from Concept Maps. *Dialogue* and *Discourse,* 3**,** 75-99.

Papasalouros, A., Kanaris, K. and Kotis, K. (2008) Automatic Generation of Multiple Choice Questions from Domain Ontologies. e-Learning, 2008. Citeseer, 427-434.

Silberschatz, A., Galvin, P.B. and Gagne, G. (2012) Operating System Concepts. New Jersey: John Wiley & Sons.

Soldatova, L.N. and Mizoguchi, R. (2009) An Ontology-Based Test Generation System. *In:* Dicheva, D., Mizoguchi, R and Greer, J. (eds.) *Semantic Web Technologies for E-Learning.* Amsterdam, Netherland: IOS Press.