Automatic Generation Of Multiple Choice Questions From Domain Ontologies.

Andreas Papasalouros
University of the Aegean
118 PUBLICATIONS **569** CITATIONS
SEE PROFILE

Konstantinos I. Kotis
University of the Aegean
74 PUBLICATIONS **746** CITATIONS
SEE PROFILE

AUTOMATIC GENERATION OF MULTIPLE CHOICE QUESTIONS FROM DOMAIN ONTOLOGIES

Andreas Papasalouros
Department of Mathematics
University of the Aegean
83200 Karlovassi, Samos, Greece

Konstantinos Kanaris, Konstantinos Kotis
Department of Information and Communication Systems Engineering
University of the Aegean
83200 Karlovassi, Samos, Greece

ABSTRACT

The aim of this paper is to present an innovative approach for generating multiple choice questions in automatic way. Although other approaches have been already reported in the literature, the approach presented in this paper is based on domain specific ontologies and it is independent of lexicons such as WordNet or other linguistic resources. The paper also reports on a first prototype implementation of such an approach which creates multiple choice question items using the Semantic Web standard technology OWL (Ontology Web Language). The proposed approach is independent of the domain since questions are generated according to specific ontology-based strategies. In current implementation, simple natural language generation techniques are used in order to provide the items in the questionnaires.

KEYWORDS

Ontologies, multiple choice questions, assessment

1. INTRODUCTION

Multiple choice questions (MCQ) are a very popular means of assessment and self-assessment in both traditional and electronic learning settings. They are appealing to the examinees, they can be automatically graded and they provide the capability of frequent testing, almost immediate feedback on the performance. An MCQ questionnaire comprises a number of questions named *items*. Each item consists of a short text describing a question or a sentence to be tested, called *stem*, and a number of alternative *choices*, typically four. In single-response MCQ, one of the choices is the *correct answer* and the wrong alternatives are called *distractors*.

Typically, the number of items constituting a questionnaire must be large enough in order for a questionnaire to provide credible evaluation. Thus, the creation of MCQ questionnaires is a time-consuming task which would be benefited by automation.

Automatic creation of questionnaires can be considered as a specialized application of natural language generation which is based on the following:

1. The existence of a knowledge base expressed in a knowledge representation language, which contains a set of facts (Bateman 1997) about a specific domain. From these facts, question items together with correct answers are extracted for the questionnaire.
2. The use of the semantic relationships between various elements in the knowledge base in order to assert 'false' sentences. These sentences are used for generating the distractors in question items.
3. The application of Natural Language Generation (NLG) techniques for actual sentence generation.

This paper describes an approach for the automatic generation of multiple choice questionnaires from domain ontologies. For experimental purposes the paper reports on results produced with a number of domain ontologies. Domain ontologies are represented in OWL format thus conforming to Semantic Web technology standards (W3C 2004). Based on this approach, a prototype tool was developed which accepts as input ontology OWL ontologies and provides as output multiple choice questionnaires. Certain strategies are used for selecting the correct answers in question items, as well for selecting the distractors. These strategies are analytically presented and constitute the main contribution of this paper. As mentioned earlier, both correct statements and distractors are converted to English sentences.

The use of ontologies in educational settings is ever-increasing. In these settings, domain specific ontologies can be provided as inputs for generating questionnaires in the following ways:

- The manual summarisation of a domain by ontology engineers and domain/ pedagogic experts, as performed in this work.
- The manual summarisation of a domain in the form of concept maps, generated either the teacher, or typically, collaboratively, by the students themselves. Concept maps can be converted to ontologies by using appropriate tools (Simón et al. 2007).
- The automatic ontology generation from text, using appropriate tools. Although this constitutes an open research problem, it is possible to create domain ontologies from text corpora such as textbooks, manuals and tutorials.
- The reuse of ontologies created for specific educational technology purposes such as educational content organisation (Boyle and Pahl 2007), searching and planning, (Dicheva and Dichev 2006), (Karampiperis and Samson 2004) and knowledge representation for intelligent tutoring systems.
- The reuse of existing domain ontologies for educational purposes. These ontologies are typically created by fiddle experts, for example, as annotations of historical archives and museum digitized resources (Trant et al. 2002). Assessment and entertainment activities can be supported by automatically generating questionnaires based on such existing ontological descriptions.

From the above it is presumed that domain ontologies are a proper formalism for providing the basis for automatic assessment.

Ontologies contain domain knowledge in the form of definitions of terms, individuals belonging to these terms and relationships between these terms and individuals. The above constitute the asserted knowledge, that is, explicitly defined facts within the ontology. Ontologies also incorporate a reasoning mechanism in order to derive facts from explicitly defined knowledge (Baader et al. 2003). These facts, not explicitly defined in the initial ontology, constitute the inferred knowledge. In this approach, reasoning is applied *before* question generation and thus, generated questions are based on both asserted and inferred knowledge. As a result, a student performing a test is assessed on recalling factual knowledge, but also is expected to apply some 'lower level intellectual skills', in the sense of simple domain specific rules, in order to answer questions based on inferred knowledge. These skills are referred by Gagné et al. (1992) as *concrete* and *defined concepts* and are related to the ability to *identify* and *classify* specific individuals as members of particular concepts. Nevertheless, domain ontologies are not capable of specifying 'procedural knowledge' and thus they cannot be used alone for assessing higher order cognitive skills (Holohan et al. 2006).

The rest of this paper is organized as follows: In Section 2 related work is presented; in Section 3 the strategies for question generation are discussed, while Section 4 describes the generation of questions and insights of the implementation of the prototype system. In Section 5 a preliminary evaluation of the approach is presented; the paper ends with some conclusions in Section 6.

2. RELATED WORK

The methodology presented by Mitkov et al. (2006) generates multiple choice questions based on text corpora in a specific domain. It utilizes several techniques such as shallow parsing, term extraction, sentence transformation and computation of semantic distance. It also employs ontologies such as WordNet (Miller 1995). The proposed system functionality can be divided in three steps: term extraction concerning frequent concepts inside the text, stem generation and distractor selection. The extraction of the terms occurs by shallow parsing of scanned text corpora. Afterwards, a frequency measure is applied so the noun or noun phrase with occurrence frequency above a customizable threshold is selected. The stem generation actually filters the clauses and transforms the selected ones to the stem of an item. This is done by utilizing a simple set of rules that are assisted by WordNet. The last part namely the distraction selection part is dictionary-based and uses mostly WordNet to obtain the candidate distractors.

In (Holohan et al. 2005) the OntoAWare system is described which provides a set of tools useful for learning content authoring, management and delivery. It exploits the semantic web technology along with knowledge-representation standards and knowledge-processing techniques. Moreover the authoring environment that is introduced by the system concerns the semi-automatic generation of the learning objects (standard e-learning and courseware elements). In order to generate the learning objects one can customize existing ontologies or even create a new one from scratch. One of the features of the presented system is the generation of questionnaires from ontology elements, which is the focus of our paper. However, the work in OntoAWare focuses on adaptivity and personalisation. The delivery environment of this system can also be configured and can vary from free navigation to learning technology standards-based course delivery in the form of Simple Sequencing specification. This version of OntAWare is focusing mainly on personalisation and not in question generation per se. As assumed by examples presented in the paper, it relies only on subsumption relationships between classes in order to generate questions while the approach presented here exploits also other kinds of structural relationships between ontology classes and individuals.

In (Holohan et al. 2006) an advancement of the OntoAWare system is presented towards generating assessments for problem solving skills in the domain of relational databases. These assessments are produced by utilizing an ontology which describes the domain in question. Students may customize the system in order to produce personalized problems.

3. STRATEGIES FOR ONTOLOGY-BASED QUESTION GENERATION

OWL is the standard Web ontology language, based on Description Logics knowledge representation formalism (Baader et al. 2003). The approach presented in this paper follows specific ontology-related conventions: *A*, *B, C*, *D* are names of *concepts* (also known as *classes)*, *R*, *S* are names of *roles* (also known as *relationships or properties)* and *a*, *b*, *c* are names of *individuals* (also known as *instances)*. Based on these conventions, we produce the following statements:

A(*a*): states that *a* is an individual of class *A*

R(*b*,*c*): states that individuals *b* and *c* participate in binary role *R.*

In the presented examples, individuals are typeset in italics, while classes and properties are typeset in sanserif fonts. Thus, *Eupalinos* and *EupalinosTunnel* are both individuals, Engineer and Tunnel are both concepts that describe the type of the related individuals respectively (i.e. *Eupalinos* is an Engineer and *EupalinosTunnel* is a Tunnel), and wasBuiltBy(*Eupalinos*, *EupalinosTunnel*) is a property that relates two individuals (binary role).

The strategies presented in this section deal only with the semantics and not with the syntactic aspects of question formation. Sentence generation is discussed in next Section. All strategies were selected so as to provide distractors semantically as similar as possible to the correct answers, so that they successfully mislead students not knowing the correct answer. In the following subsections, these strategies are presented together with examples taken from a domain ontology in the Greek ancient history domain called 'Eupalinos Tunnel', which is illustrated in Figure 1. Strategies are distinguished into three major categories, depending on the elements in the ontology that are used to extract the appropriate knowledge for sentence creation.

3.1 Class-based Strategies

In ontologies, individuals are members of collections named concepts or classes. These classes are organised in subsumption (class/subclass) hierarchies, that is, 'is-a' relationships as depicted in Figure 1. This category contains strategies that generate distractors based on classes and their individuals. For all strategies, correct answers are of the following type: 'Instance *a* is a Class A', e.g. '*Eupalinos* is an Engineer'. In ontology engineering terms, the above sentence means that *Eupalinos* is an instance of Engineer or that *Eupalinos* is of type Engineer.

Distractors are formed by creating sentences in the same format as the correct answer, by choosing proper individuals or classes different than those that appear in the correct answer.

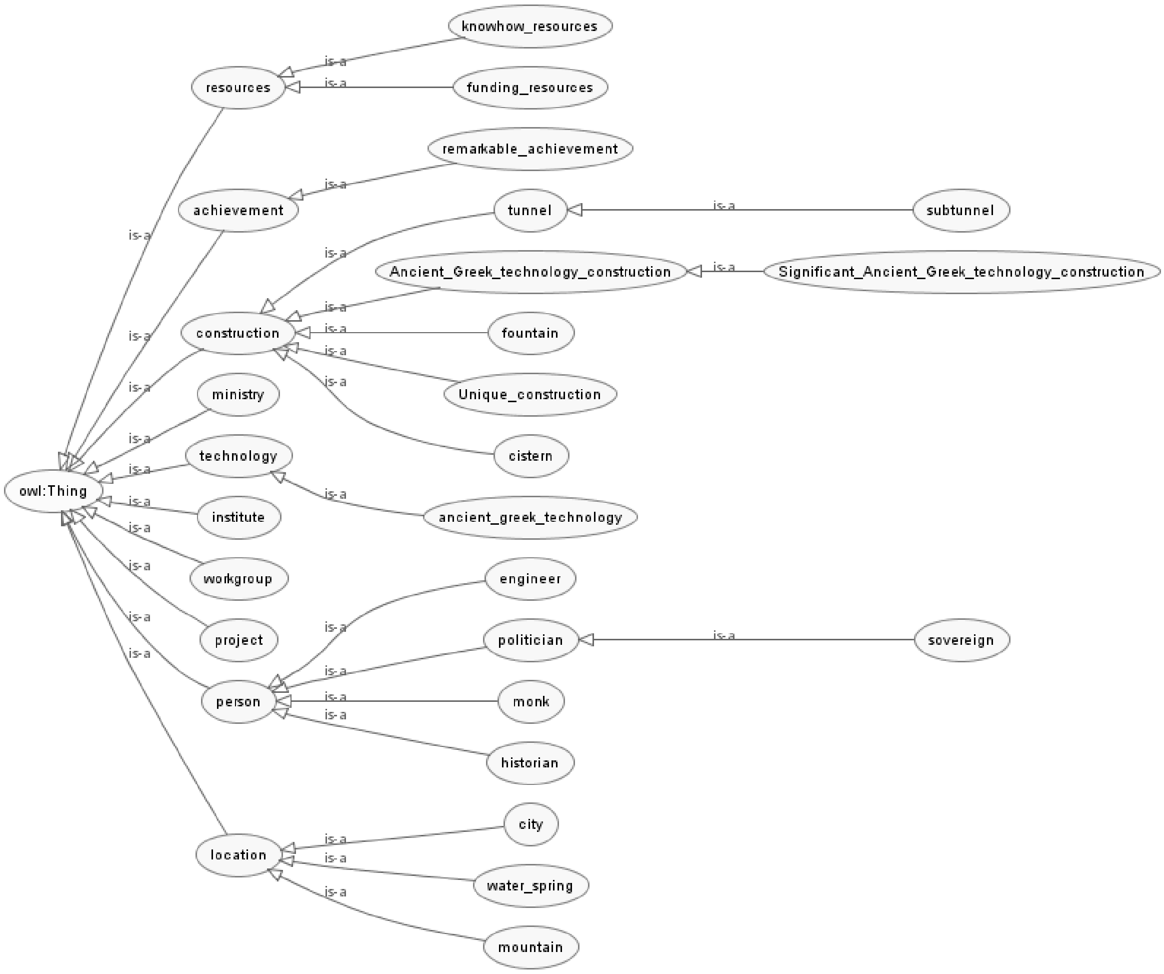


Figure 1. Sample ontology Eupalinos Tunnel

Strategy 1. Choose individuals which are not members of a given class, provided that they are members of one of its superclasses. More specifically, if *A(a)* for some *a,* then correct answer is: *A(a)*. For the distractors selection, we assume that *B* is a superclass of *A*. Then, if *B(b)*, *b≠a* and *b* is not an individual of *A*, then *A(b)* is a distractor.

Example: '*Ampelos Hill* is a Mountain' is the correct answer, since *Ampelos Hill* is an instance of class Mountain. "*North Opening of main tunnel* is a Mountain" is a distractor, since *North Opening of main tunnel* is an instance of concept Location, which is a superclass of class Mountain. As shown in the example, distractors formed by this strategy differ from the correct answer in the name of individual used as subject.

Strategy 2. Choose individuals belonging to disjoint siblings of a given class in order to generate distractors. If *A(a), b≠a*, *B(b)* and *B* sibling class (disjoint or not) of *A* then a distractor is: *A(b)*. Note that if *B* is not a disjoint sibling of *A*, the strategy still applies, but in this case there is a possibility that a distractor is a correct answer, which leads to an invalid distractor.

Example: '*Eupalinos* is an Engineer' is the correct answer, since *Eupalinos* is an instance of class Engineer. '*Polykrates* is an Engineer' is a distractor, since *Polykrates* is an instance of concept Politician and Engineer and Politician are disjoint subconcepts of class Person.

Strategy 3. Choose individuals belonging to a class that has a non empty intersection with a given class. If *A(a),* there exist individuals *b* and *c* such as *c≠a*, *A(b), B(b)* and *B(c)* then *A*(*c*) is a distractor.

Example: '*Eupalinian Aqueduct* is a Remarkable Achievement' is the correct answer, since *Eupalinian Aqueduct* is an instance of class Remarkable Achievement. '*Dyros* is a Remarkable Achievement' is a distractor, since *Dyros* is an instance of class Tunnel and *Eupalinian Aqueduct* is a member of both classes Tunnel and Remarkable Achievement (Tunnel and Remarkable Achievement have a non empty intersection).

Strategy 4. Choose sibling classes (disjoint or not) to a given class. If *A(a), B* is a sibling of *A*, disjoint or not, then *B*(*a*) is a distractor. Distractors differ from correct answers in the name of the class. This strategy is dual to strategy 2.

Example: '*Eupalinos* is an Engineer' is the correct answer, since *Eupalinos* is an instance of class Engineer. '*Eupalinos* is a Politician' is the distractor, since Engineer and Politician are disjoint subclasses of class Person.

Strategy 5. Choose subclasses of a given class. If *A(a)* and, *B* is a superclass of *A* and *a* is not a member of *B* then *B*(*a*) is a distractor. Again, distractors differ from correct answers in the name of the class in generated sentence. This strategy is dual to strategy 1.

Example: '*Aristrachus* is a Person' is the correct answer, since *Aristrachus* is a member of class Person. '*Aristarchus* is an Engineer' is a distractor since Engineer is a subclass of Person.

3.2 Property-based Strategies

This category contains strategies that create question items and distractors based on properties (roles), that is, relationships between individuals in the ontology. A property has a *domain*, which is the "class of individuals to which this property can be applied" and a *range*, which is the "class of individuals that a property can have as its value" (W3C 2004). There are two kinds of properties in OWL: object properties, which are relationships between individuals and datatype properties, that is, relationships between individuals and basic types, e.g. numerical or string. In terms of OWL, *R* is an *object property* and *b*, *c* are individuals which are related by this particular property. Correct answers are generated from property instances in the ontology, *R*(*a*,*b*), that is, individuals *a*,*b* related with property *R*.

Strategy 6. Choose individuals from a class equal or subclass of the domain of a given property. If *, R(a,b), c ≠ a* and *c* an individual of a class which is equal to or subclass of the domain of property *R* then *R(c,b)* is a distractor.

Example: '*Polykrates* hired *Eupalinos'* is the correct answer, since the domain of this property is class Person. '*Herodotus* hired *Eupalinos'* is a distractor, since *Herodotus* is an instance of class Historian which is a subclass of class Person.

Strategy 7. Choose individual members of a class which is equal or subclass of the range of a given property to generate distractors. If property *R, R(a,b), c ≠ b* and *c* an individual of a class equal or subclass of the range of property *R* then *R(a,c)* is a distractor.

Example: '*Polykrates* hired *Eupalinos'* is the correct answer, since the range of this property is class Person. '*Polykrates* hired *Herodotus'* is a distractor, since *Herodotus* is an instance of class Historian which is a subclass of class Person (the property range).

Strategy 8. Choose a property having both domain and range equal or subclass of the domain and range of the property of the correct answer. More formally, if a property *S* has a domain and a range that are equal or subset to the domain and range of property *R* correspondingly then if *R(a,b)* is a correct answer then *S(a,b)* is a distractor.

Example: '*Eupalinian Aqueduct* brings water to *ancient city of Samos'* is the correct answer. '*Eupalinian Aqueduct* leads to *ancient city of Samos'* is a distractor, since property leads to has range Location and domain Tunnel.

Strategy 9. Choose a numeric *datatype property* value by taking multiples and submultiples of a given property value. This strategy is based on numeric datatype properties, that is, on properties that relate individuals to numerical values.

Example: If '*Eupalinian Aqueduct* years spent for completion 10' is the correct answer, then '*Eupalinian Aqueduct* years spent for completion 16' is a distractor.

3.3 Terminology-based Strategies

Strategies in this category are based on concept/subconcept relationships, without dealing with ontology individuals at all.

Strategy 10. Choose sibling classes of the direct superclass of a given class to substitute the subject of the correct answer. If class *A* is subclass of *B*, *B* is a direct subclass of *C*, *D* is another direct subclass of *C* then a distractor is: *D is a B*. Distractors differ from correct answers in the class used as subject.

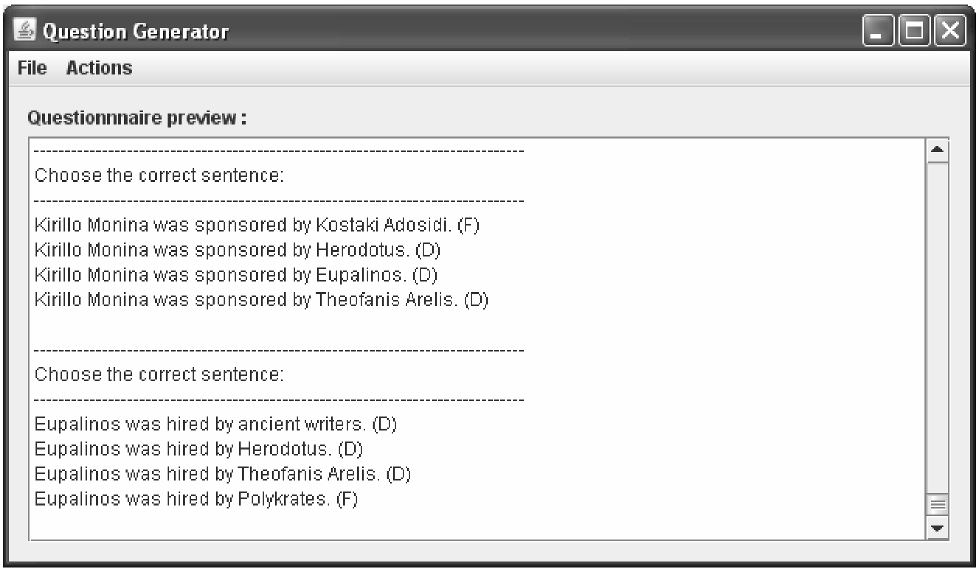
Example: ‘Sovereigns are politicians’ is the correct answer, since Sovereign is a subclass of Politician. ‘Monks are politicians’ is a distractor, since Monk is a sibling class of Politician.

Strategy 11. Choose sibling classes of the direct superclass of a given class to substitute the object of the correct answer. If class *A* is subclass of *B*, *B* is a subclass of *C*, *D* is another subclass of *C* then a distractor is: *A* is a *D*. Distractors differ from correct answers in the class used as object.

Example: ‘Sovereigns are politicians’ is the correct answer, since Sovereign is a subclass of Politician. ‘Sovereigns are monks’ is a distractor, since Monk is a sibling class of Politician.

4. MULTIPLE CHOICE QUESTION GENERATION PROTOTYPE

A prototype tool was developed which accepts as input OWL documents and generates questionnaires in plain text. The format of questions is illustrated in Figure 2.



```
--------------------------------------------------------------------------------
Choose the correct sentence:
--------------------------------------------------------------------------------
Kirillo Monina was sponsored by Kostaki Adosidi. (F)
Kirillo Monina was sponsored by Herodotus. (D)
Kirillo Monina was sponsored by Eupalinos. (D)
Kirillo Monina was sponsored by Theofanis Arelis. (D)

--------------------------------------------------------------------------------
Choose the correct sentence:
--------------------------------------------------------------------------------
Eupalinos was hired by ancient writers. (D)
Eupalinos was hired by Herodotus. (D)
Eupalinos was hired by Theofanis Arelis. (D)
Eupalinos was hired by Polykrates. (F)
```

Figure 2. The MCQ generation system output. (F) and (D) symbols have been added only for presentation/evaluation reasons and they are not part of the actual system output.

In all types of questions, the stem is ‘Choose the correct sentence:’. In question items based on instance-of relationships, that is, for class-based strategies, declarations in the form Concept(*Individual*) in the ontology are transformed in sentences of the form ‘*Individual* is a(n) Class’. As an example, ‘Eupalinos is an engineer’. For strategies based on subclass relationships, that is, for questions generated by terminology-based strategies, the form generated strategies is illustrated in the examples of strategies no 10 and 11 above, that is, class names appear in plural number. Alternative forms of sentences for the above types of questions can be defined using Hearst patterns (Hearst 1992). For questions generated by propert-based strategies, sentences are generated in the form ‘*Individual* propertyName *Individual*’. As illustrated in Figure 2, was_sponsored_by is the name of an object property and thus, an example of a generated sentence is ‘Kirillo Monina was sponsored by Kostaki Adosidi’. The name of the property is tokenized according to simple rules, for example, underscore is recognized as a separating character. Sentence generation is performed using the SimpleNLG natural language generation framework (http://www.csd.abdn.ac.uk/~ereiter/simplenlg/).

The MCQ application was implemented in Java. JENA Semantic Web framework (http://jena.sourceforge.net/index.html) was used for OWL ontologies management and storing and thus for the implementation of strategies presented in previous section. Pellet open source DL reasoner (http://pellet.owldl.com/) was used as an inference engine, that is, for subsumption and automatic classification of individuals in the ontology.

5. EVALUATION

Five ontologies from different domains were used for evaluating the proposed approach. Some metrics pertaining to the number of classes, individuals and properties contained in these ontologies are presented in Table 1. From the example ontologies, *Eupalineio Tunnel* ontology was developed by domain experts and ontology engineers as a test-case for the method presented in this paper.

Table 1. Example ontologies used for evaluating the proposed approach.

	Eupalineio Tunnel	*MSc Program*	*Travel v.1*	*Travel v.2*	*Grid Resources*
Individuals	40	100	145	38	21
Classes	29	25	72	21	54
Object prop.	25	38	22	13	26
Datatype prop.	16	23	13	0	24

The generated questionnaires were evaluated in three dimensions: Pedagogical quality, linguistic/ syntactical correctness and number of questions produced. These dimensions were considered for each strategy category.

Table 2. Multiple choice question items per question generation strategy

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*Total*
Eupalineio T.	3/3	5/8	5/5	5/8	1/2	32/41	6/6	3/3	0/6	3/3	3/3	66/88
Travel 2		5/6		3/5		0/17				4/4	4/4	16/36
Travel 1		3/17		5/14		1/3	0/39		0/55	19/19	19/19	47/166
MSc Progr		3/6		0/5					0/59			3/70
Grid Res.				0/1					0/1	6/6	6/6	12/14

The generated questions from the Eupalineio Tunnel ontology were reviewed by two domain/educational experts. All questions were found satisfactory for assessment. Nevertheless, all questions are not syntactically correct. Table 2 depicts the number of syntactically correct items and the total number of questions per strategy for each ontology. In order to overcome syntactic problems, more sophisticated natural language generation techniques should be utilized in future implementations of the presented approach.

Input ontologies should adhere to certain conventions in order to generate syntactically correct sentences, with the NLG techniques adopted so far. The most important is that properties' names must be written as verbs or verb-like phrases.

Property-based strategies may produce a large number of multiple choice questions but are very difficult to manipulate syntactically. Class and terminology-based strategies on the other hand are much easier to handle syntactically but generate fewer questions for ontologies of the same depth and population.

While the proposed approach works well in defining the semantics of questions, the problem of generating syntactically correct question items is only partially tackled.

6. CONCLUSION

In this paper, a new approach for automatic generation of multiple choice questions has been presented. The proposed approach is based on strategies that use ontological axioms and asserted/inferred knowledge of a knowledge base developed in OWL. A prototype tool has been developed for evaluation reasons, proving that the proposed approach, when used with semantically-rich and fully populated domain ontologies, can provide successful cases.

Future work on this direction includes the evaluation of the approach with elementary school students, the identification of further strategies (e.g. ones that involve multimedia objects, supporting multimedia standards and ontologies) as well as improvements of the NLG subsystem. Furthermore, initial experiments have also been started in order to provide an automatic mechanism (no human intervention) for enriching the domain ontologies either in the level of concepts/properties (ontology learning) or in the level of individuals (ontology population). Such a way, the proposed approach will also handle ontologies that cannot utilize the proposed strategies due to lack of domain knowledge. Our first experiments on this direction include information sources such as the Web, using search engines such as Google to ‘fish’ individuals from on-line lexicons or other linguistic resources (Googling).

ACKNOWLEDGEMENT

This work was partly supported by the EU-IST-034567 Grid4All project. We thank Professor George A. Vouros for his comments and suggestions on this work. We also thank the domain experts and school teachers involved in the evaluation process as well as in the refinement of the experimental ontologies. Especially, we thank the archeology/museum studies/educational programs’ specialist Mrs. Stamatia Ladikou, the pre-school teacher and educational programs’ specialist Mrs Aikaterini Koti for their expertise on multiple choice questions tests.

REFERENCES

Baader, F. et al, 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, USA.

Bateman, J., 1997, Enabling technology for multilingual natural language generation: the KPML development environment, *Natural Language Engineering*, Vol. 3, pp. 15-55.

Boyce, S. and Pahl, C., 2007, Developing Domain Ontologies for Course Content. *Educational Technology & Society*, Vol. 10, No. 3, pp. 275-288.

Dicheva, D. and Dichev, C., 2006. TM4L: Creating and Browsing Educational Topic Maps, *British Journal of Educational Technology*, Vol. 37, No. 3, pp. 391-404.

Gagné et al, 1992, *Principles of Instructional designd,* 4th edn, Harcourt Brace, New York.

Hearst, M., 1992. Automatic acquisition of hyponyms from large text corpora. *Proceedings of COLING*. pp. 539-545.

Holohan, E. et al, 2005. Adaptive e-learning content generation based on semantic web technology. *Proceedings of Workshop on Applications of Semantic Web Technologies for e-Learning*, Amsterdam, The Netherlands, pp. 29-36.

Holohan, E. et al, 2006. The Generation of E-Learning Exercise Problems from Subject Ontologies. *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies,* Kerkrade, The Netherlands, pp. 967-969.

Karampiperis, P. and Sampson, D., 2004, Adaptive instructional planning using ontologies. *Proceedings of the Forth IEEE International Conference on Advanced Learning Technologies*, Joensuu, Finland pp. 126-130.

Miller, G. et al, 1995. Five papers on WordNet. *CSL Report 43*, Cognitive Science Laboratory, Princeton University.

Mitkov, R. et al, 2006. A computer-aided environment for generating multiple-choice test items. *Natural Language Engineering,* Vol. 12, No. 2, pp. 177-194.

Simón, A. et al, 2007. Generation of OWL Ontologies from Concept Maps in Shallow Domains, in D. Borrajo, L. Castillo, and J.M. Corchado (Eds.): *CAEPIA 2007, LNAI 4788*, pp. 259-267.

Trant, J. and Bearman, D., 2002. Open concepts: museum digital documentation for education through the AMICO Library, *Visual Resources Association Bulletin,* http://www.amico.org/docs/papers/2002/ArtDoc0302.pdf [Accessed April 14th 2008].

W3C, 2004. OWL Web Ontology Language Overview, W3C Recommendation, http://www.w3.org/TR/owl-features/ [Accessed April 14th 2008].