EVALUATION OF DOMAIN ONTOLOGY REQUIREMENTS IN E-TUTOR FRAMEWORK

Ghanim Hussein Ali Ahmed
PhD student, University of Miskolc, Institute of Information Science
3515 Miskolc, Miskolc-Egyetemváros

László Kovács
professor, University of Miskolc, Institute of Information Science
3515 Miskolc, Miskolc-Egyetemváros, e-mail: kovacs@iit.uni-miskolc.hu

Abstract
The domain module within Intelligent Tutoring Systems is used to describe the course topics including the knowledge units and competencies. Ontology is a good solution for knowledge management as it can provide a standardized way and a reasoning engine for the related knowledge management tasks. In this paper, we analyze the functionality of the usual OWL-based ontology construction and set the main requirements to enhance the usability of the ontology knowledge modelling language. Based on these shortcomings, we set the main requirements against an efficient ITS domain ontology description language. This language should be based on simplified Manchester syntax and it should contain additional structural elements too.

Keywords: *Intelligent Tutoring System, ontology, knowledge modeling*

1. Introduction

Intelligent Tutoring Systems (ITS) have a common goal of enabling learning in a meaningful and effective manner by using a variety of computing technologies. There are many examples of ITS being used in both formal education and professional settings in which they have demonstrated their capabilities and limitations. An ITS typically aims to replicate the demonstrated benefits of one-to-one, personalized tutoring in contexts where students would otherwise have access to one-to-many instruction from a single teacher (e.g., classroom lectures), or no teacher at all (e.g., online homework) [1]. ITS are often designed with the goal of providing access to high quality education to each and every student.

The domain module within the ITS is used to describe the course topics including the knowledge units and competencies. In the literature there are many candidate knowledge representation formats. Ontology is a good solution for knowledge management as it can provide a standardized way and a reasoning engine for the related knowledge management tasks. In this paper, we analyze the functionality of the usual OWL-based ontology construction and set the main requirements to enhance the usability of the ontology knowledge modelling language.

2. Intelligent e-tutor frameworks

Intelligent Tutoring System is a computer system that aims to provide immediate and customized instruction or feedback to learners, usually without requiring intervention from a human teacher. Researchers, designers, and developers define ITS in different ways. According to Fletcher and Sottilare [2], intelligent tutoring may be viewed as “an effort to capture in computer technology the capabilities and practices of a human instructor who is expert in both the subject matter and one-to-one tutoring”. From the earliest days of computers, researchers have struggled to develop intelligent tutoring systems that are as effective as human tutors [3].

In education, AI promises to deliver what has always been the highest goal of pedagogy, wise and caring guidance for each student, adapted to the individual’s needs. AI would potentially provide customized learning resources and activities, combined with the pace and style of instruction that best suits each individual student [4]. As a result, all learners would have their own highly intelligent digital tutor [5] and the problem of personalized education could be solved once and for all. The remaining teachers would supervise and adjust this computational adaptive learning [6].

There is a close relationship between intelligent tutoring, cognitive learning theories and design; and there is ongoing research to improve the effectiveness of ITS. An ITS typically aims to replicate the demonstrated benefits of one-to-one, personalized tutoring, in contexts where students would otherwise have access to one-to-many instruction from a single teacher (e.g., classroom lectures), or no teacher at all (e.g., online homework) [7].

Intelligent Tutoring Systems (ITS) got much attention from researchers even though ITS educational technology began in the late 1960s. ITSs are tutors based on computers that serve as a supplement to human teachers. The main benefits of an ITS is that it can provide learners with personalized guidelines based on their cognitive skills. ITS is a computer-based training system that incorporate techniques for communicating / transferring knowledge and skills to students. ITS is a combination of Computer-Aided Instruction (CAI) and Artificial Intelligence (AI) technology. Hausmann et al. [8] defined ITS as a computer science and cognitive science to create computerized tutoring systems that offer immediate feedback and individualized instruction.

3. Ontology models

Current ITS systems are based on some key functional components. The systems usually contain the following modules (Fig 1):

- domain module to describe the course topics including the knowledge units and competencies,
- student module to model the behaviour of the students,
- teacher module to model the behaviour of the tutor,
- course repository to store the different study aids and course materials,
- interface module to implement intelligent, human oriented interface to the system.

This structure can be observed among others in Intelligent Tutoring System Builder (ITSB)[9] or in Cognitive Tutor Authoring Tools (CTAT) [10].

The domain module should provide an efficient knowledge management in ITS. Regarding the different knowledge management tools [11], we can summarize the main requirements in the following points:

- extensibility: the model can be extended with new knowledge units on an easy way,
- reusability: the model can be re-used in many different application areas, it should provide a tool for model integration,

- efficient operations: the base knowledge management operations (like knowledge retrieval, reasoning, validation) should be executed in an acceptable time,
- flexibility: it should incorporate knowledge units from many different areas, application domains,
- standardized: the interface and storage should use a standard model,
- supporting model validation: the constructed model should support integrity rules and it should provide tools to perform model validation.

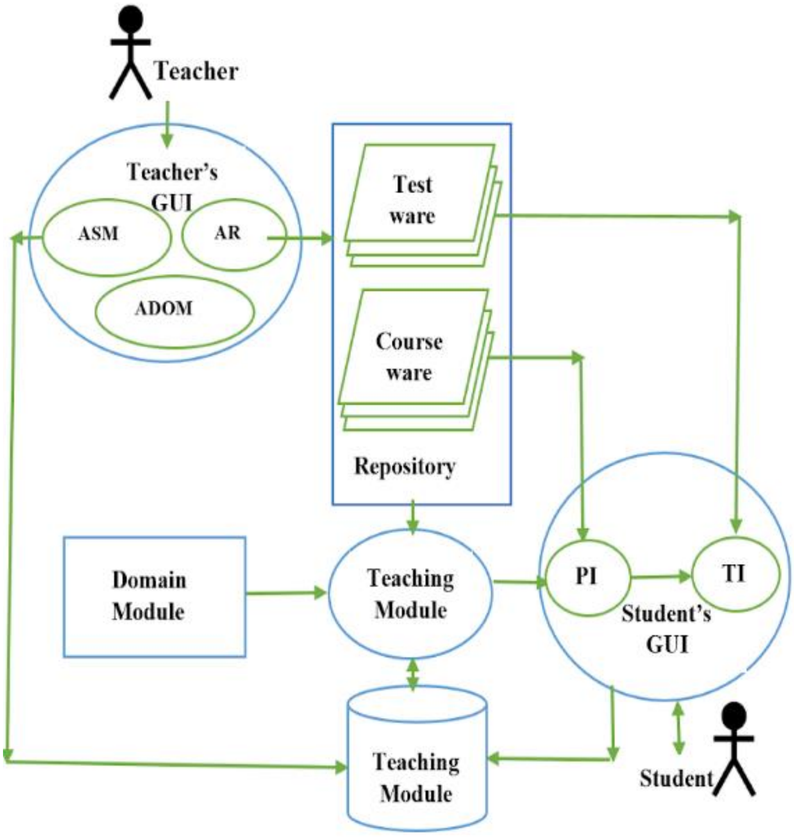

Fig 1, ITS architecture [9]

In order to meet these requirements, the ITS systems usually use ontology representation formalism for knowledge management. Ontology can be defined as an explicit conceptualisation in IT application using a standard rich representation model and a reasoning engine for logic operations [12]. Current ontology models are based either on RDF or on the OWL standards. In OWL-DL, the following components can be used to construct an ontology model [13]:

- concepts, classes
- individuals
- object properties
- data properties
- class axioms (equivalence, subclass, disjoint)

- Description logic (first order logic + cardinality constraints, domain constraints) to define complex, derived classes
- SWRL implication rule systems to generate derived facts
- SPARQL or DL Query to retrieve base or derived facts
- The ontology database can be implemented at
  - file level, using separate OWL/RDF data files
  - server level using standard databases or RDF triplet stores

The server can be accessed either with standard program API or with REST interfaces. A good survey on comparison of the different implementation methods can be found among others in [14].

4. Domain ontology

There are many efforts to standardize the semantic description of documents or of application domains. The model should identify the main content units and the key relationship among the content units. One of the first general document content models is defined in the DITA standard [15]. DITA model consists of three main building blocks: topics, maps and output formats. The topic is used to present a comprehensive information unit. There are three different subtypes of topic: concept topic, task topic and reference topic describing background information. The topic element may include different components to describe the activity phases, like prerequisite, context, sections or steps. The map and classification elements can describe any kind of relationships among the topic elements including hasKind or hasPart.

The DITA model is used generally as a baseline model for other model extensions. For example, in [16], a simplified version was presented, where the categorization of the content elements is based primarily on the representation format of the content unit.

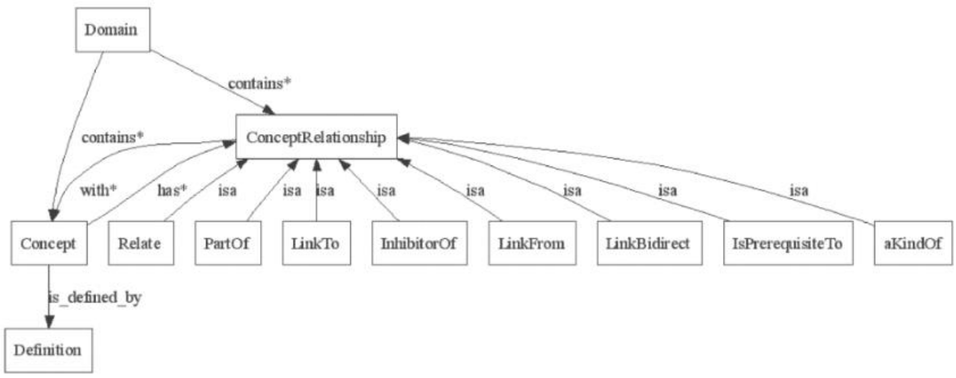

Fig 2. Domain model schema[16]

Regarding the special requirements of the ITS systems, we set the following objectives at the beginning of our research to construct an ITS domain model:

- identification of the different knowledge units and association links,
- providing information about the handling of the knowledge units,
- automatic processing (question generation, question answering, hint generation, assessment evaluation) of the knowledge units,

- link to external knowledge databases,
- standardized representation format,
- extendable metadata about the knowledge units.

Considering the methodological aspects, we have selected the programmed learning approach [17] as educational method to be applied in the ITS system. According to [18], programmed learning is the process of arranging the material to be learned into a series of sequential steps, where these programmes can be used with many types of students and subject matter by themselves; in the name of self-instruction or in combination with other instructional techniques.

Based on the suggestions in the related literature on knowledge management [19] and on education methodology [20], we propose the following key elements in the domain model:

- Topic (Class)
- TopicContent (Class)
- Frame (Class); subclasses:
  - Motivation (Class)
  - Explanation (Class)
  - Term Definition (Class)
  - Process Definition (Class)
  - Evaluation (Class)
  - Application (Class)
  - Illustration (Class)
  - Exercises, demonstration (Class)
  - Micro-test (Class)
  - Assessment (Class)
  - Question / Correct answer (Class)
- ControlFlow (Class); subclasses:
  - Sequencing (Class)
  - Branching (Class)
  - Loop (Class)
  - Frame (Class)
- Relationship (Object Property); subclasses:
  - Prerequisite (Object Property)
  - Containment (Object Property)
  - Specialization (Object Property)
  - Media source (Object Property)
- Teaching material (Class); subclasses:
  - Text source (Class)
  - Video source (Class)
  - External URI (Class)
- Attribute (Data Property):
  - Objectives and competencies (Data Property)
  - Complexity level (Data Property)

- ◦ Methodology (Data Property)
- ◦ Condition (Data Property)
- ◦ Result (Data Property)
- ◦ Key words (Data Property)

The presented domain model can be converted into a T-Box [21] OWL ontology framework on a straightforward way using the following relationships.

ontology element	concept model element
object property Prerequisite	Frame => Frame; Frame => ExternalURI
object property Containment	Topic => TopicContent; TopicContent => ControlFlow; Sequencing =>> ControlFlow; Branching => (Sequencing,Sequencing); Loop => Sequencing; Assessment =>> Question / Correct answer; MictroTest =>> Question / Correct answer
object property MediaSource	Frame =>>Text source; Frame =>> VideoSource; Frame =>> ExternalURI
data property Objectives and competencies	Topic => String; Frame => String
data property Complexity level	Topic => String; Frame => String
data property Condition	Branching => Expr; Loop => Expr;
data property Result	Micro-test => Integer; Assessment => Integer; Question / Correct answer => Integer

Table 1. Domain model in T-Box format

5. Experiences with the ontology construction

We have tested the proposed syntax with several smaller examples. In Fig 2, we present an ontology fragment in Protege where the Topic object relates to the SQL INSERT statement. The prototype system was constructed with the Protege ontology editor and the model is stored in OWL/XML format. The TopicContent element contains a sequence, where the first element of the sequence is a motivation element with a text source describing the operations on relational tables. Next step is a micro-test verifying the knowledge on SQL INSERT command. If the test fails then an explanation element is executed presenting the syntax for the INSERT command.

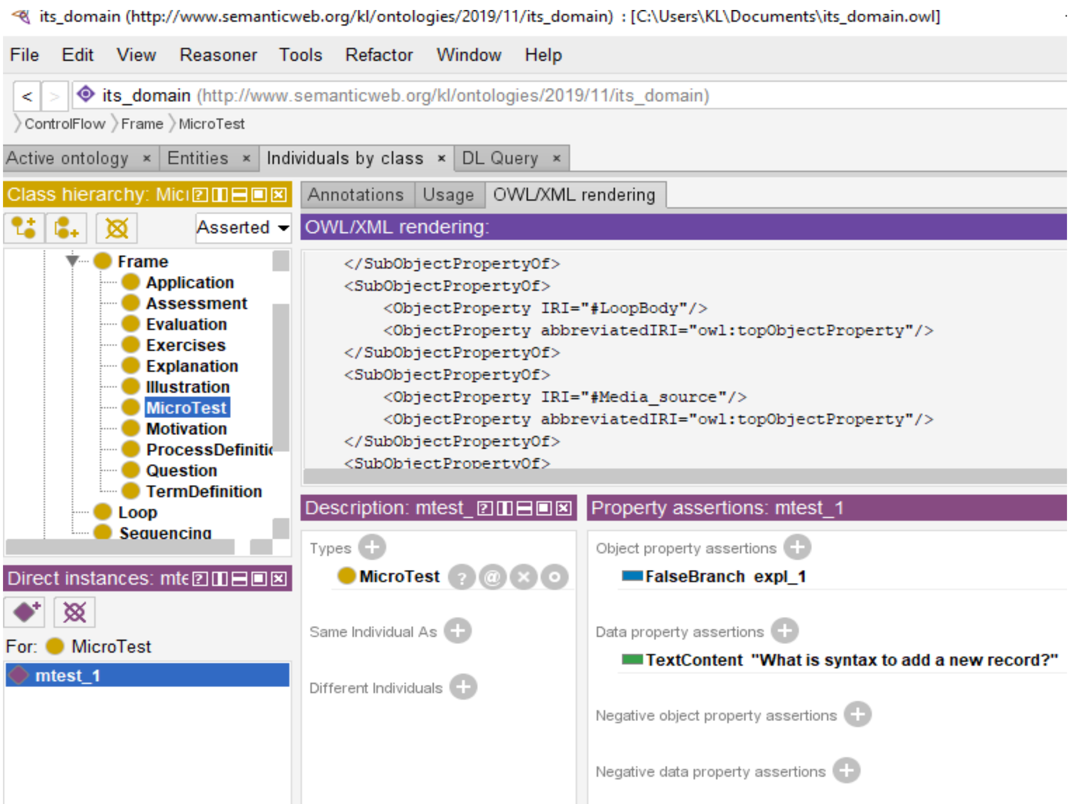

Fig 3. Ontology editor

Regarding the ontology construction, the direct ontology construction with Protege has some drawbacks which should be eliminated in the next phase of the development process. Our experiences can be summarized in the following points:

- The resulted OWL ontology file is very long, cannot be processed by manual way. The ontology file should be used only as a background data store. The Manchester Syntax which is a simplified version of the WL XML syntax [22], provides a more compact format but,
- It may contain redundancy
- It does not cover all necessary structural elements

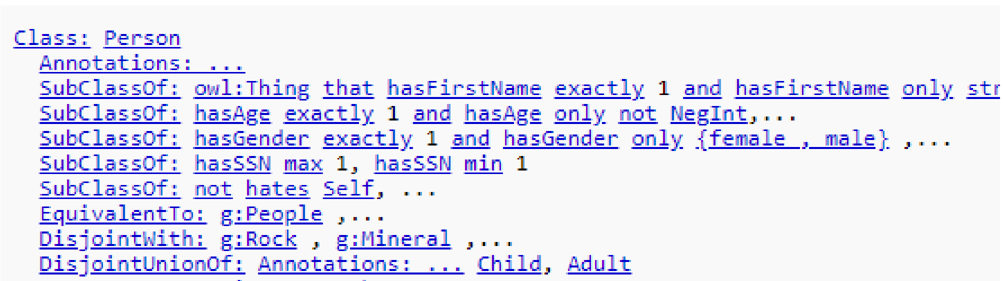

Fig 4. Ontology database schema

A specific modelling language should be used to construct the knowledge model instead of the direct ontology OWL syntax. A good example for the language difficulty is the sequence construction. The domain model contains several sequence structures while the OWL has no direct element for this important unit. In the literature, we can find some approaches how to mimic the list structure in OWL-DL, but these solutions seem to be very complex for an efficient implementation.

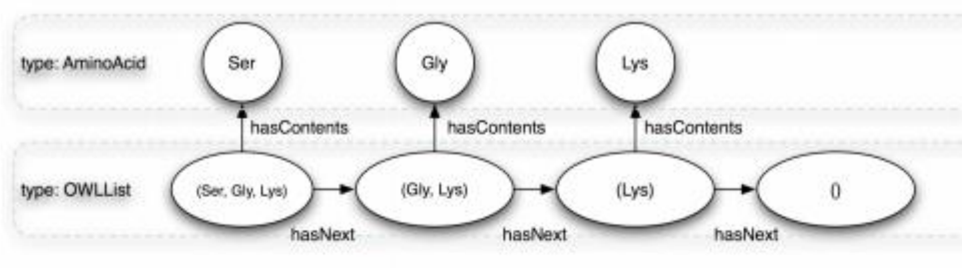

Fig 5. Sequence construction example

The description language should support the DL and SWRL syntax for an efficient representation of the class derivation and rule description components.

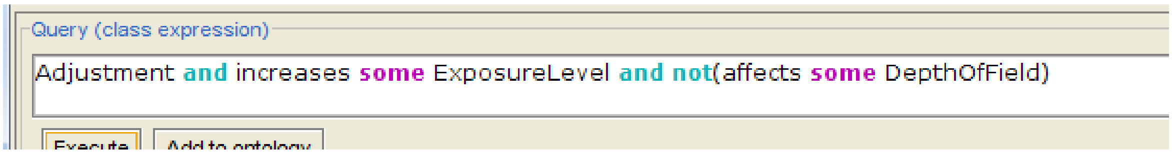

Fig 6. SWRL rule example

6. Conclusion

The domain model is a key component in the ITS systems. The domain model describes the knowledge and competency elements together with the processing workflow of the training sessions. Ontology can provide a standardized way and a reasoning engine for the related knowledge management. Based on our experiences, current ontology technologies lack of simplicity and structural functionality. Based on these shortcomings, we set the main requirements against an efficient ITS domain ontology description language. This language should be based on simplified Manchester syntax and it should contain additional structural elements too.

Acknowledgements

The research reported here was carried out as part of the EFOP-3.6.1-16-2016-00011 "Younger and Renewing University – Innovative Knowledge City – Institutional development of the University of Miskolc aiming at intelligent specialization" project implemented in the framework of the Széchenyi 2020 program. The realization of this project is supported by the European Union, co-financed by the European Social Fund.

References

[1] Freedman, Reva; Ali , S. S.; Mcroy, S. What is an intelligent tutoring system. Intelligence, 2000, 11(3)9:15-16. **https://doi.org/10.1145/350752.350756**

[2] Robert Sottilare, et al, Design Recommendations for Intelligent Tutoring Systems, 2015, Florida.: U.S. Army Research Laboratory.

[3] Nkambou, Roger; Mizoguchi, Riichiro; Bourdeau, Jacqueline (ed.), Advances in intelligent tutoring systems, 2010, Springer Science & Business Media, **https://doi.org/10.1007/978-3-642-14363-2**

[4] Burns, H., Luckhardt, C. A., Parlett, J. W., & Redfield, C. L. (Eds.), Intelligent tutoring systems: Evolutions in design, 2014, Psychology Press. **https://doi.org/10.4324/9781315807492**

[5] Moen, H. C., & Moolman, D. , U.S. Patent Application No. 13/144,350,2012.

[6] Brusilovsky, P., & Millán, E. . User models for adaptive hypermedia and adaptive educational systems. The adaptive web 2007:3-53 **https://doi.org/10.1007/978-3-540-72079-9_1**

[7] Hong, C. M., Chen, C. M., Chang, M. H., Chen, S. C. Intelligent web-based tutoring system with personalized learning path guidance. In Seventh IEEE International Conference on Advanced Learning Technologies, 2007, pp. 512-516 **https://doi.org/10.1109/ICALT.2007.167**

[8] Grignetti, M. C., Hausmann, C., & Gould, L. An intelligent on-line assistant and tutor: NLS-SCHOLAR. In Proceedings of the May 19-22, 1975, National computer conference and exposition, pp. 775-781 **https://doi.org/10.1145/1499949.1500121**

[9] S. S. A. Naser., ITSB: An Intelligent Tutoring System Authoring Tool., Journal of Scientific and Engineering Research. Research Article (2016).

[10] G. G. S. J. a. C. Z. Chris Feng, Authoring Tools for Easy Creation of Adaptive Tutoring., BHCI Capstone Project. Spring, 2018.

[11] Tang, A., Avgeriou, P., Jansen, A., Capilla, R., Babar, M. A., A comparative study of architecture knowledge management tools. Journal of Systems and Software 2010, 83(3):352-370. **https://doi.org/10.1016/j.jss.2009.08.032**

[12] Gruber, T. Ontology , 2009, pp. 1963-1965 **https://doi.org/10.1007/978-0-387-39940-9_1318**

[13] Pan, J. Z., & Thomas, E. Approximating owl-dl ontologies. In Proceedings of the National Conference on Artificial Intelligence, 2007, pp. 1434.

[14] Jeong, D., Choi, M., Jeon, Y. S., Han, Y. H., Yang, L. T., Jeong, Y. S., & Han, S. K., Persistent storage system for efficient management of OWL web ontology. In International Conference on Ubiquitous Intelligence and Computing, 2007, pp. 1089-1097 **https://doi.org/10.1007/978-3-540-73549-6_106**

[15] Jovanovic, J., Gasevic, D., & Devedzic, V. Ontology-based automatic annotation of learning content. International Journal on Semantic Web and Information Systems 2006, 2(2):91-119. **https://doi.org/10.4018/jswis.2006040103**

[16] Nešić, S. Semantic document model to enhance data and knowledge interoperability. Web 2.0 & Semantic Web (2010), pp. 135-160 **https://doi.org/10.1007/978-1-4419-1219-0_6**

[17] Chastain, K., Behavioristic and cognitive approaches in programmed instruction. Language learning, 1970, 20(2):223-235. **https://doi.org/10.1111/j.1467-1770.1970.tb00479.x**

[18] Smith, W., & Moore, J. W. , Size-of-step and achievement in programmed spelling. Psychological Reports 1962, 10(1): 287-294. **https://doi.org/10.2466/pr0.1962.10.1.287**

[19] Liebowitz, J., & Frank, M. (Eds.), Knowledge management and e-learning., 2016, CRC press. **https://doi.org/10.1201/b10347**

[20] Zellweger, F. , Faculty adoption of educational technology. EDUCAUSE quarterly 2007, 30(1):66-69.

[21] Serafini, L., Borgida, A., & Tamilin, A. Aspects of distributed and modular ontology reasoning, IJCAI 2005, 5:570-575.

[22] Horridge, M., Drummond, N., Goodwin, J., Rector, A. L., Stevens, R., & Wang, H. The Manchester OWL syntax. OWLed 2006, 216