Development of Ontology-based Domain Knowledge Model for IT Domain in e-Tutor Systems

Ghanim Hussein Ali Ahmed, Jawad Alshboul, László Kovács
University of Miskolc, Faculty of Mechanical Engineering and Informatics, Miskolc, Hungary

***Abstract*—Ontology as a technology has been studied in many areas and is being used in several fields. A number of studies have utilized ontology to manage problems such as interoperability in teaching materials, modeling and enriching education resources, and personalizing learning content recommendations in the educational context. A possible reason for the lack of success may be that simply posting lecture notes on the internet does not provide enough learning and training. However, this situation can improve by using education software like an e-tutoring system. The e-tutor system has built-in modules to track students' performance and personalize learning according to an adaptation of students' learning styles, knowledge levels, and proper teaching techniques in e-learning systems. e-Tutor is an excellent area in the context of electronic instruction since it provides adequate aid for learners and becomes increasingly important for individual and collaborative learning. Thus, there has been significant interest in adopting e-tutoring to facilitate learning processes and enhance learners' performance. This paper represents a domain knowledge model for an e-tutoring system that enables knowledge to be stored in such a way separated from the domain of interest and assists in storing transfer and prerequisite knowledge relationships. This innovative technique is helpful for the students in improving their learning progress. This paper introduced a domain knowledge model for an e-tutor system to support the way of teaching and learning process. The model implementation is developed in Python and Owlready2. Two types of ontologies are provided: general concepts of the domain knowledge ontology and specific domain knowledge ontology. This solution represents the knowledge to be learned, delivers input to the expert model, and eventually provides specific feedback, selects problems, generates guidance, and supports the student model.**

***Keywords*—e-learning; knowledge model; domain model; e-tutor system; SPARQL**

I. Introduction

E-learning is a technique of learning and teaching utilizing Information and Communication Technology. It means that the growth of online learning is a collaborative task including several researchers in several disciplines such as educational design and learning material [1]. e-Learning system is frequently getting popular in the academic community due to the advantages of using it anywhere, and anytime [2]. However, it tends to be most commonly employed for web-based education to access online courses via the internet [2]. A possible reason for the lack of success may be that merely posting lecture notes on the internet does not provide enough learning and training [2]. This situation can be improved by using education software such as an e-tutoring system. The e-tutoring system includes built-in modules to monitor a learner's performance and personalize education based on adaptation to learners' learning style, existing knowledge level, and suitable teaching approaches in e-learning systems. The e-tutoring framework usually applies to computer-based instruction. However, an e-tutoring system generally uses adaptive mechanisms and concepts to solve the current e-learning problems by applying artificial intelligence techniques.

Artificial Intelligence in Education (AIED) society frequently identifies the value of producing technologies with a global reach [3]. e-Tutor system is a computer application that uses artificial intelligence approaches to improve and personalize the teaching and learning processes. e-Tutoring differs from other e-learning systems because it uses a knowledge base to guide the pedagogical approach. It attempts to optimize the student's mastery of domain knowledge by generating new problems, concepts, and instruction feedback. e-Tutoring systems are computer-based teaching systems that regularly give students quick and personalized guidance or feedback without mediation from a personal teacher. e-Tutoring is an excellent subject in the context of e-learning. e-Learning environments are increasingly getting popular in different contexts in academies, universities, and vocational training [4]. Therefore, suitable support of learners is also getting great significance. Besides, collaborative learning is also growing, which puts greater demands on learners, especially when the collaboration is implemented and tailored to enhance learning and learning results [4].

The area of instruction is one of the first areas in which ontologies are applied in as a cognitive tool. For several considerations, this was a consequence to the extensive adoption of the constructivist paradigm of education and the general application of such knowledge technologies as concept maps, mind maps, and several more for learning goals [5]. Recently, scientists found that the benefits of using ontologies in the educational field have relative support for designing the coming e-learning environment. The term “ontology” comes from philosophy and it is defined as “a set of representational primitives with which to model a domain of knowledge or discourse” [6]. So, an ontology gives a specific view on some part of the world. While knowledge representation formalisms define how to represent concepts, ontologies identify the concepts to be described and are linked. Thus, ontology can be noticed as a well-established and widely agreed-upon system of concepts in a specific knowledge domain and their relationship. Specialized knowledge domain ontologies can be utilized as a semantic pillar for topics or repositories of teaching resources. By offering agreed-upon terminologies for the domain knowledge representation, ontologies would help share, reuse, and exchange topic units. Ontologies also enable the availability of machine-readable web resources.

Most of the current solutions of e-tutor systems are designed and developed for a particular domain which means the provided solution will not be suitable for another knowledge domain. Therefore, these systems developed for isolated knowledge bases have some limitations and drawbacks for using local knowledge bases. These limitations refer to lack of standardization, limited knowledge base shareability, lack of flexibility, lack of reusability, and manual control. Due to these shortcomings, an ontology domain knowledge model is proposed to avoid the limitations and drawbacks of using local knowledge bases. To improve and increase the learning quality and process, a novel ontology domain knowledge model develops by defining a set of relationships that would be adequate and clear to represent all possible relationships for developing and building the ontology domain knowledge model. In the proposed ontology domain knowledge model, two types of domain ontologies were introduced: a) general concepts for domain knowledge model ontology and b) specific domain knowledge model ontology. The general concepts of the proposed ontology domain knowledge model deal with the domain knowledge model concepts and define the relationship related to these concepts. The ontology of a selected domain knowledge model deals with the selected subject area that can relate the selected subject domain to the general concepts for the domain knowledge model. It seems like individuals or instances for the general concepts of the domain knowledge model.

Another key issue in current systems is the limited concept level functionality. Many systems provide data-level features as they lack the information and engine to perform smart operations. The proposed ontology framework includes reasoning engines that can improve the adaptivity, customization of the e-tutor frameworks. In the proposed knowledge base, the schema can contain also rules as part of the background ontology. This approach supports a declarative description model instead of the procedural way; thus, it has an increased flexibility and coding efficiency.

This paper constructs a domain knowledge model for an e-tutor system which in turn would help enhancing teaching and learning process. Furthermore, the implementation of this model is described in Python, which can be applied in the future to support the problem-solving process.

This article is organized as follows. The introduction and methodology of designing the ontology are explained in the first and second sections. The third section displays the related work, the fourth section illustrates the proposed ontology of the domain knowledge model and explains the proposal model in detail, the fifth section presents a case study on ontology implementation using Python and Owlready2 module, the sixth section demonstrates the result and discussion, and the conclusion is in the seventh section.

II. METHODOLOGY

The primary goal of this work is to enhance the quality of the learning process by making it in a personalized way. The process design and development of ontology usually encompasses several standard tasks. However, there is no dominating approach for constructing the ontologies. The main principle is to define the ontology concepts which are related to the objects and the relationships for the selected domain. The methodology for creating and building an ontology assumes defining the objectives and domain of applicability. Moreover, it must be identified in higher-level details: what is the purpose of designing the domain ontology, what are the types of questions that should be answered through it, how it will be utilized and supported for solving the problem for the selected domain. Several techniques for the design and development of ontology are given such as [7] and [8]. Though these methods are somewhat different and are influenced in varying ways by the technology used, the underlying processes of developing the domain ontology are similar. Therefore, the suggested ontology development process is composed of the following phases:

- Domain and purpose of the ontology.
- Discover if there are related ontologies.
- Enumerate important terms in the domain.
- Defining the key classes and their hierarchy.
- Identify the properties of classes.
- Facets attaching to properties.
- Creating class instances.

III. RELATED WORK

The technique known as e-tutoring system has been tracked by scholars in education, psychology, and artificial intelligence. The aim of e-tutoring system is to offer the advantages of one-to-one teaching. It allows students to train their skills by bringing out activities in greatly interactive learning platforms. e-Tutoring system is a computer application that uses artificial intelligence approaches to improve and personalize the teaching and learning processes [9]. e-Tutoring differs from other e-learning systems because it uses a knowledge base to guide the pedagogical approach. It attempts to optimize the student's mastery of domain knowledge by generating new problems, concepts, and instruction feedback. There are many e-tutoring systems meant and developed for learning and teaching reasons. These systems support learners to progress quickly and improve their self-confidence. SmartTutor is a web-based intelligent tutoring system developed to support teaching and students based on knowledge background, skills, and teaching techniques [10]. The remarkable aspect of SmartTutor is the incorporating of instructional and artificial intelligence (AI) methods in a unified intelligent e-tutoring system to give personalized help to learners based on the knowledge-level of individual learners [11]. e-Tutoring systems are developed to illustrate the essential knowledge on the subject, inform which kinds of knowledge can be used to solve problems in the given domain, generate and offer suitable task activity based on the recorded performance, and recommend the next suitable task for the learner to choose. Smart Tutor delivers the capability of a tutor to adjust to individual learners' requirements and skills [10]. e-Tutoring systems, also known as Intelligent Tutoring Systems

(ITS), became a reality in the mid-1970s and peaked at the end of the 1980s, when expert systems were in use [12]. During the nineties, most artificial intelligence and education investigations concentrated on intelligent learning environments more influenced by Computer-based learning, microworlds, and Computer-based training [13]. Of course, ITS study is still an active sub-field of research on adaptive learning. ITS systems were not cost-effective enough to survive in education and training. ITS got interested in the late 2010s, particularly since "statistical AI" based on neural networks and relevant methods allows extracting patterns from big data. These can be due to recommender systems that recommend learning analytic systems that detect learners' difficulties. ITS are adaptive systems that use intelligent technologies to tailor learning based on various characteristics of students, including their background knowledge, mood, and learning style. The e-tutoring system includes three kinds of knowledge to give students proper education, arranged into four separate system modules (as shown in Fig. 1).

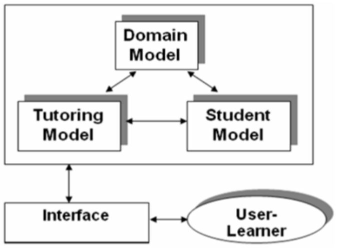


Fig. 1. Traditional Components of an ITS [10].

The tutor model is a computer model that helps the system choose the appropriate explanation style for a particular learner based on the learner’s history tracked by the learner model. This knowledge allows the e-tutoring system to check the student's responses and steps with an expert to assess what they know and do not know. The student model is the learner's knowledge-level while the student works on e-tutoring system. The model assesses student's behavior while working with the e-tutoring system to find their knowledge, cognitive skills, and reasoning abilities. The domain model is concerned with the knowledge associated with the problem considered for a specific domain, including the teaching materials and the meta-information about the topic to be taught. It depends on the learner model's diagnostic processes to determine what, when, and how to display information to a learner. The interface model is essential as a delivery mechanism and learning facility to help the learner complete the activity. It can also perform as an external representation of the expert and teaching models.

Researchers have investigated practices of knowledge representation such as semantic-based, rule-based, case-based, frame-based, Bayesian network, logic-based, and ontology-based. Rule-based models are also called Cognitive tutors. The rule-based models are built from cognitive task analysis, producing problem spaces or task models. These problem spaces or task models are constructed by observing the expert and novice users. Task models represent a set of production rules in which each rule represents an action corresponding to a task [14]. When a user tries to solve a given task, the user's reasoning ability is analyzed based on the rules applied by the user, i.e., the user's solution is compared step-by-step to the solution given by the expert. Case-based is an artificial intelligence problem-solving method that records experience into cases and associates the current problem with an experience [15]. A logical representation language has some definite rules for dealing with propositions and reasoning for knowledge and representation. Logical representation entails deducing a conclusion from many circumstances. This representation establishes many fundamental communication principles. It is composed of well-defined syntax and semantics that facilitate sound inference. Each phrase can transform into a logical form through syntax and semantics. Frame-Based is one of the artificial intelligence techniques to structure the data, and it is used to separate information into substructures via the representation of stereotyped scenarios [16]. Frame-Based seems like a record form build-up of many characteristics and values used to describe an object in the real world. In addition, this object contains a collection of slots and their associated values. However, these slots come in a combination of shapes and sizes. Facets are the names and values assigned to slots. A Bayesian network is also referred to as a belief network or Bayes net. Bayesian network is probabilistic and graphical in a form of graph with directed acyclic devoid of loops and self-connections used for knowledge representation for an uncertain domain, with each node indicating a random variable [17]. Each edge denotes a conditional probability associated with the associated random variables. Semantic-based is a knowledge representation method that enables visualization of the knowledge via graphical networks [18]. This network incorporates nodes representing entities and arcs that reflect their relationships. Moreover, Semantic-based classify things in different ways and can also connect them in the style of a graph.

Ontologies, as semantic-based representation, have gained vital significance as one of the most commonly used techniques to describe and share knowledge in several disciplines such as E-learning systems, business modeling, software engineering, knowledge engineering [19].

Regarding the continuous development of new technology, it can change the way of teaching and learning. According to Fensel [20], the primary reason for the popularity of ontologies is due to providing “a shared and common understanding of a domain that can be communicated between people and application systems. Ontology can be constructed as a representation required for scale and variety in the design of educational frameworks. In the e-learning field, ontologies are employed in various applications extending from domain knowledge modules representation to automate generation and assessment of personalized learning materials. The concept of ontology is a useful technology that incorporates related resources, shares knowledge, and eliminates unnecessary data. Ontology is a fundamental description of the information in the world [21]. The ontology in computing refers to knowledge representation applying a collection of concepts and connections among them [22]. In the context of a targeted discipline, ontology is used to rationally reason and validate concepts in the semantic knowledge model. In theory, ontology is a "formal, explicit specification of a shared conceptualization" [23]. It offers a shared vocabulary that can be employed to construct the domain knowledge model, involving objects, concepts, properties, and relationships.

A comparison was given utilizing some selected criteria according to the knowledge representation models employed in the present works for representing the domain knowledge module as a part of an E-tutor framework. The authors considered some criteria for comparing the representation model used with others in the literature. The suggested criteria covered a number of terminologies: standardbility, reusability, flexibility, shareability, simplicity, and reasoning engine, as indicated in Table I.

TABLE I. COMPARISON OF THE MODELS

Models	**Criteria**					
	Standard ability	***Reusab ility***	***Flexibi lity***	***Share ability***	***Simpli city***	***Reason ing engine***
Semantic-based	√	√	×	√	×	√
Rule-based	√	√	×	×	×	√
Case-based	√	×	×	×	×	√
Frame-based	√	√	√	×	×	√
Bayesian network	√	×	×	×	×	√
Logic-based	√	×	×	√	×	√
Ontology-based	√	√	√	√	√	√

√ means feature is allowed, and × means feature not allowed.

IV. PROPOSED DOMAIN KNOWLEDGE MODEL FOR E-TUTOR FRAMEWORK

The domain model is the system's knowledge base, and it organizes the domain knowledge structure, its various key concepts, and the relationships between the concepts. This model essentially deals with the what-to-teach part of e-tutoring system [24]. The domain model is concerned with domain knowledge construction, organization, topics, and relationships [24]. Domain knowledge is a set of suggestions that identify all the vocabulary concepts to explain or solve problems. Domain knowledge is only declarative, and it does not tell how learners can use the domain knowledge model to solve a practical problem [25].

Based on the properties of the learning materials, two kinds of ontologies are employed, and these are general concepts domain knowledge ontology and specific domain knowledge ontology. These modules represent the knowledge to be learned, deliver input to the expert model, and eventually provide specific feedback, select problems, generate guidance, and support the student model.

A novel domain knowledge model was suggested based on the current research area, as shown in Fig. 2. This model is based on topics, concepts, attributes, tasks, competencies, assessment, and relations. In order to share and reuse the domain knowledge model in e-tutoring systems, ontologies are employed to organize and represent the domain knowledge model. The benefit of this model is to personalize the materials for learners.

Based on the general concepts of the domain knowledge ontology shown in Fig. 2, topics, concepts, attributes, tasks, competencies, and assessment terms refer to the following:

- Topics can be utilized to present domain knowledge or a comprehensive overview of a subject or course.
- The concept identifies the sub-domain or unit of a subject or course.
- Competency is used to demonstrate the features and skills that allow and enhance the efficiency of student performance to gain new knowledge and understand specific topics.
- The task is used to demonstrate how a student can complete a task within a given period of time.
- The attribute represents a topic or domain attribute within a domain model.
- The assessment is used to present how the system can evaluate or assess the student activities required within a given period of time.

Fig. 3 displays the design of a specific domain knowledge ontology case study for IT domain in e-tutor system for the computer programming. Many types of relationships are used in the selected case study, such as specialization or generalization, association, and containment. A containment means that a specific topic within a domain contains different concepts (has-a). The specialization or generalization means that certain topics or domains have specific concepts (is-a). The association means that a specific topic or concepts associate with each other. Based on Fig. 2 and Fig. 3, the following shows a brief description of a subject:

- Topic: Control Structure.
- Concept: Loop, Sequence and Condition.
- Competency: understand, analyze, implement.
- Task: program, code review, project.
- Attribute: syntax, operators.
- Assessment: activities such as quizzes, tests.

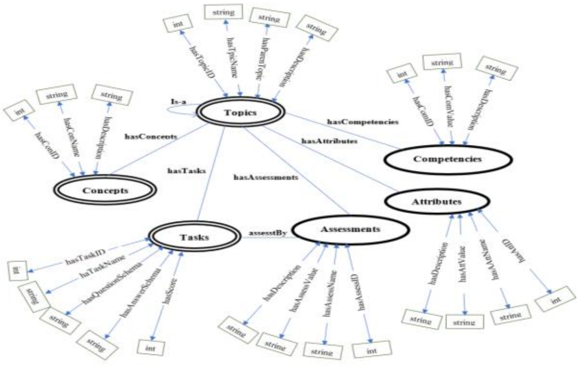


Fig. 2. The Proposed Ontology-based Model for Domain Knowledge.

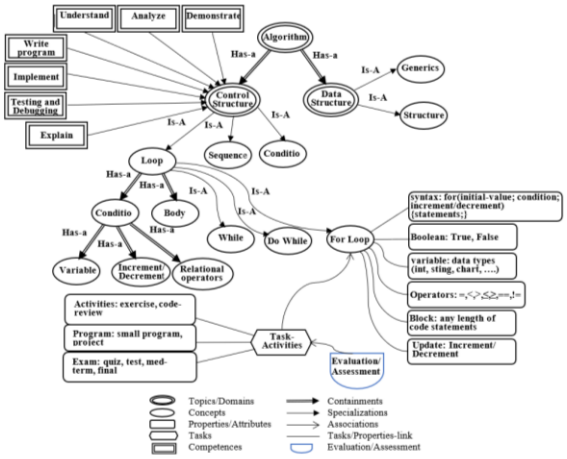
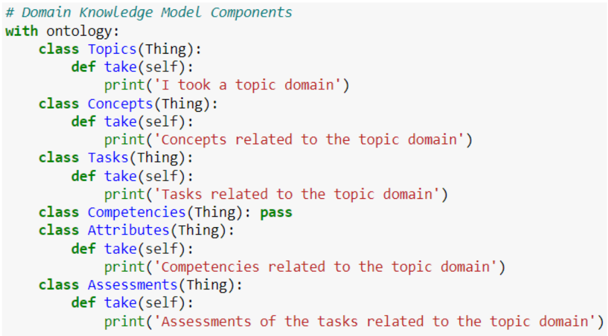


Fig. 3. Domain Knowledge Module Instances for e-tutoring Frameworks.

V. Implementation of the Proposed Model

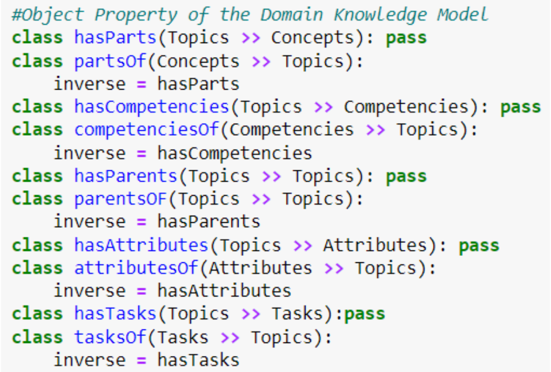
Information science and technology provides many modules and packages for ontology construction and management. Python is one of the most popular languages adopted when implementing an ontology for the domain knowledge model. It is an interpreted, object-oriented, extensible programming language [23], which provides an excellent combination of clarity and versatility in different disciplines. The domain knowledge model, considered here, is “Basics of Computer Programing”, and the ontology created consisted of the “Algorithm in Computer Programming”. Fig. 4, 5, 6, 7, 8, 9, and 10 show the implementation of the proposed model using Python.

Fig. 4 displays a snippet of the domain knowledge components construction using Python and Owlready2 syntax "with ontology: ..." to indicate that the constructed ontology will receive the new RDF triples and the class keyword for declaring the components of the ontology. While Fig. 7. demonstrates the output of the ontology of the domain knowledge model components as a list of concepts. Fig. 5. displays a snippet of the object property related to the domain knowledge model in the format of domain and range. For example, "hasParts(Topics >> Concepts)" hasParts is the object property relationship that means each Topic may contain some Concept in this example, Topics is the domain of the relationship, and Concepts is the range of the relationship. Fig. 6 displays a snippet of the data property relationship related to the domain knowledge model in the structure of domain and range. Considering "topicName(Topics >> str, FunctionalProperty)," topicName declares the name of the Topics component, Topics are the domain, and str (means the topicName datatype is a string) is the range of the relationship. Fig. 8. presents how to insert the " While Loop" topic data as an instance related to the Topics component. Checking the consistency of the ontology using the HermiT reasoner, the most commonly used in ontology engineering applying sync_reasoner(), is shown in Fig. 9. Rule construction in SWRL layouts for inferring further knowledge can add to the created domain knowledge model of the selected ontology components using imp = Imp() and imp.set_as_rule() functions, as shown in Fig. 10.



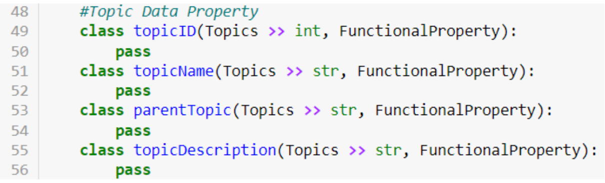
```
# Domain Knowledge Model Components
with ontology:
    class Topics(Thing):
        def take(self):
            print('I took a topic domain')
    class Concepts(Thing):
        def take(self):
            print('Concepts related to the topic domain')
    class Tasks(Thing):
        def take(self):
            print('Tasks related to the topic domain')
    class Competencies(Thing): pass
    class Attributes(Thing):
        def take(self):
            print('Competencies related to the topic domain')
    class Assessments(Thing):
        def take(self):
            print('Assessments of the tasks related to the topic domain')
```

Fig. 4. Domain Knowledge Components Construction.



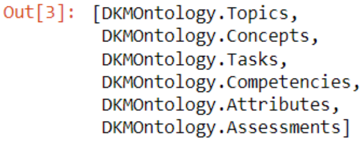
```
#Object Property of the Domain Knowledge Model
class hasParts(Topics >> Concepts): pass
class partsOf(Concepts >> Topics):
    inverse = hasParts
class hasCompetencies(Topics >> Competencies): pass
class competenciesOf(Competencies >> Topics):
    inverse = hasCompetencies
class hasParents(Topics >> Topics): pass
class parentsOF(Topics >> Topics):
    inverse = hasParents
class hasAttributes(Topics >> Attributes): pass
class attributesOf(Attributes >> Topics):
    inverse = hasAttributes
class hasTasks(Topics >> Tasks):pass
class tasksOf(Tasks >> Topics):
    inverse = hasTasks
```

Fig. 5. Object Property of the Domain Knowledge.



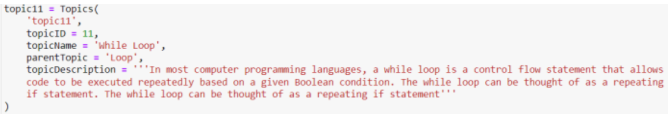
```
    #Topic Data Property
    class topicID(Topics >> int, FunctionalProperty):
        pass
    class topicName(Topics >> str, FunctionalProperty):
        pass
    class parentTopic(Topics >> str, FunctionalProperty):
        pass
    class topicDescription(Topics >> str, FunctionalProperty):
        pass
```

Fig. 6. Data Property of the Domain Knowledge.



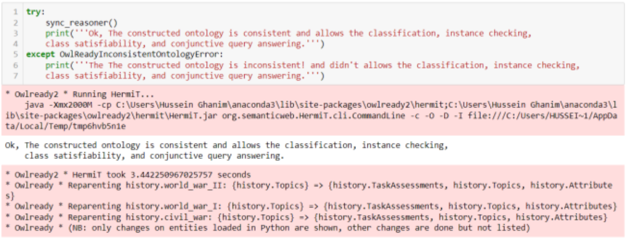
```
Out[3]: [DKMOntology.Topics,
         DKMOntology.Concepts,
         DKMOntology.Tasks,
         DKMOntology.Competencies,
         DKMOntology.Attributes,
         DKMOntology.Assessments]
```

Fig. 7. The List of Concepts.



```
topic11 = Topics(
    'topic11',
    topicID = 11,
    topicName = 'While Loop',
    parentTopic = 'Loop',
    topicDescription = '''In most computer programming languages, a while loop is a control flow statement that allows
    code to be executed repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating
    if statement. The while loop can be thought of as a repeating if statement'''
)
```

Fig. 8. A Topic Instance for While Loop.



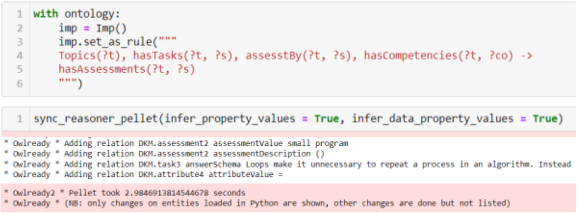
```
try:
    sync_reasoner()
    print('''Ok, The constructed ontology is consistent and allows the classification, instance checking,
    class satisfiability, and conjunctive query answering.''')
except OwlReadyInconsistentOntologyError:
    print('''The The constructed ontology is inconsistent! and didn't allows the classification, instance checking,
    class satisfiability, and conjunctive query answering.''')
```

```
* Owlready2 * Running HermiT...
    java -Xmx2000M -cp C:\Users\Hussein Ghanim\anaconda3\lib\site-packages\owlready2\hermit;C:\Users\Hussein Ghanim\anaconda3\l
ib\site-packages\owlready2\hermit\HermiT.jar org.semanticweb.HermiT.cli.CommandLine -c -O -D -I file:///C:/Users/HUSSEI~1/AppDa
ta/Local/Temp/tmp6hvb5n1e
```

```
Ok, The constructed ontology is consistent and allows the classification, instance checking,
    class satisfiability, and conjunctive query answering.
```

```
* Owlready2 * HermiT took 3.442250967025757 seconds
* Owlready * Reparenting history.world_war_II: {history.Topics} => {history.TaskAssessments, history.Topics, history.Attribute
s}
* Owlready * Reparenting history.world_war_I: {history.Topics} => {history.TaskAssessments, history.Topics, history.Attributes}
* Owlready * Reparenting history.civil_war: {history.Topics} => {history.TaskAssessments, history.Topics, history.Attributes}
* Owlready * (NB: only changes on entities loaded in Python are shown, other changes are done but not listed)
```

Fig. 9. Checking the Consistency of the Ontology.



```
with ontology:
    imp = Imp()
    imp.set_as_rule("""
    Topics(?t), hasTasks(?t, ?s), assesstBy(?t, ?s), hasCompetencies(?t, ?co) ->
    hasAssessments(?t, ?s)
    """)
```

```
sync_reasoner_pellet(infer_property_values = True, infer_data_property_values = True)
```

```
* Owlready * Adding relation DKM.assessment2 assessmentValue small program
* Owlready * Adding relation DKM.assessment2 assessmentDescription ()
* Owlready * Adding relation DKM.task3 answerSchema Loops make it unnecessary to repeat a process in an algorithm. Instead
* Owlready * Adding relation DKM.attribute4 attributeValue =

* Owlready2 * Pellet took 2.9846913814544678 seconds
* Owlready * (NB: only changes on entities loaded in Python are shown, other changes are done but not listed)
```

Fig. 10. Rule Construction and the Results of Adding New Knowledge.

Protégé is used to display the ontology graph of while loop topic shown in Fig. 11. A SPARQL query about displaying all the Topics in the developed ontology domain knowledge model, considering retrieving the topic “While” and its description is demonstrated in Fig. 12.

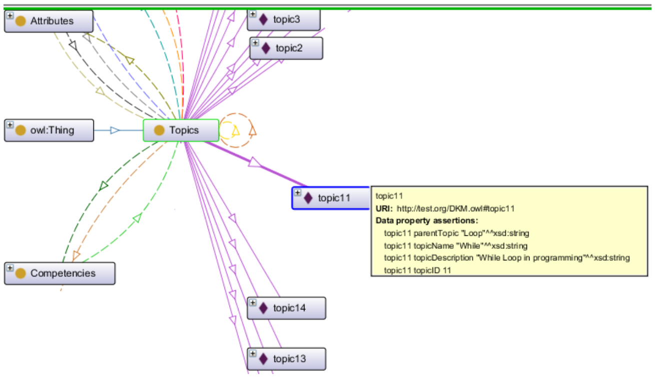
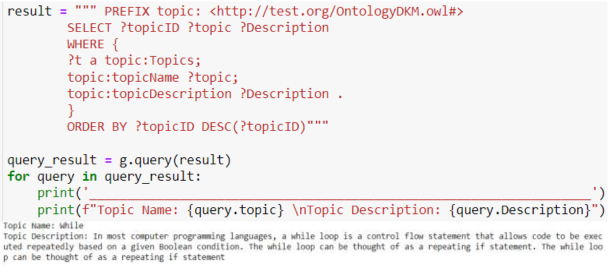

Attributes
topic3
topic2
owl:Thing
Topics
topic11
topic11
URI: http://test.org/DKM.owl#topic11
Data property assertions:
topic11 parentTopic "Loop"^^xsd:string
topic11 topicName "While"^^xsd:string
topic11 topicDescription "While Loop in programming"^^xsd:string
topic11 topicID 11
Competencies
topic14
topic13

Fig. 11. The Ontology Graph of While Loop Topic.



```
result = """ PREFIX topic: <http://test.org/OntologyDKM.owl#>
        SELECT ?topicID ?topic ?Description
        WHERE {
        ?t a topic:Topics;
        topic:topicName ?topic;
        topic:topicDescription ?Description .
        }
        ORDER BY ?topicID DESC(?topicID)"""

query_result = g.query(result)
for query in query_result:
    print('_______________________________________________________________________________')
    print(f"Topic Name: {query.topic} \nTopic Description: {query.Description}")
Topic Name: While
Topic Description: In most computer programming languages, a while loop is a control flow statement that allows code to be exec
uted repeatedly based on a given Boolean condition. The while loop can be thought of as a repeating if statement. The while loo
p can be thought of as a repeating if statement
```

Fig. 12. A SPARQL Query to Retrieve the Topic “While” and its Description.

VI. Result and Discussion

The most commonly utilized knowledge representations are semantic-based, rule-based, case-based, frame-based, Bayesian network, logic-based, and ontology-based [14], [15], [16], [17], [18] approaches. A comparison of knowledge representation forms is conducted according to some criteria or features covered: standardbility, reusability, flexibility, shareability, simplicity, and reasoning engine. Ontologies, as knowledge-based representation, have gained vital significance as one of the most commonly used techniques to describe and share knowledge in several disciplines such as E-learning systems, business modeling, software engineering, knowledge engineering [19].

The semantic web is an extension of the World Wide Web, is crucial in the development of personalized learning in e-learning systems. Semantic web technologies such as RDF, XML, and ontologies can be used for knowledge representation and reasoning of the teaching materials. Ontologies, which can be defined as a specification of a conceptualization, have the advantage of addressing the challenges of interoperability between educational repositories of various e-learning systems.

An ontology-based domain knowledge model has been proposed. In section 4, a theoretical model is described based on two kinds of ontologies. First, a general concepts domain knowledge ontology based on topics, concepts, attributes, tasks, competencies, assessment, and relations is presented in Fig. 2. Second, a specific domain knowledge ontology is designed as a case study for IT domain in programming using different types of relationships such as specialization or generalization, association, and containment as depicted in Fig. 3. In Section 4, an implementation of the ontology-based model is delivered using Python, Owlready2, and Protégé.

The current work deals with declarative knowledge that helps students understand all the vocabulary concepts in a specific subject. The future work suggests dealing with problem-solving process that works on procedural knowledge to help students in understanding how to use domain knowledge in solving a practical problem.

The proposed model is introduced to explain how the ontology domain knowledge model can be combined with an e-tutor system to improve the quality of intelligent problem-solving. Also, it will make it possible to reuse knowledge components and develop e-tutor system frameworks. Finally, a domain knowledge model for e-tutoring system, which can serve to enhance teaching and learning, is proposed. Furthermore, this model can avoid the issue currently exists in intelligent e-learning systems related to isolating knowledge bases. The majority of the domain knowledge models use an isolated knowledge base, and this local knowledge base can only provide limited knowledge background. The limitations of the isolated knowledge base are lack of standardability, reusability, flexibility, and limited knowledge. The solution can satisfy the characteristics of reusability, standardability, open knowledge, and flexibility.

By using ontology as a knowledge representation technique for building the domain knowledge model, the problem of isolated knowledge bases can be avoided. The developed ontology can be involved to manage adaptive intelligent e-learning frameworks in the future.

VII. Conclusion

An intelligent tutoring system or E-tutoring System is a type of educational system that uses artificial intelligence to reflect knowledge. The use of e-tutoring systems has become a key component of enhancing educational activities. In this paper, readers can see that e-tutoring is very significant in the context of the growing number of e-learning frameworks. An ontology-based knowledge model for IT domain has been developed for e-tutor system.

This paper constructs a domain knowledge model for an e-tutor system which in turn would help enhancing teaching and learning process. Furthermore, the implementation of this model is described in Python and Owlready2 module, which can be applied in the future to support the problem-solving process. Two types of ontologies are employed, and these are general concepts domain knowledge ontology and specific domain knowledge ontology. These modules represent the knowledge to be learned, deliver input to the expert model, and eventually provide specific feedback, select problems, generate guidance, and support the student model.

The current work deals with declarative knowledge that helps students understand all the vocabulary concepts in a specific subject. The future work suggests dealing with problem-solving process that works on procedural knowledge to help students in understanding how to use domain knowledge in solving a practical problem.

Acknowledgment

The described study was carried out as part of the EFOP-3.6.1-16-00011 "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 Szechenyi 2020 program. The realization of this project is supported by the European Union, co-financed by the European Social Fund.

References

[1] R. Prakash and S. Kumar, "E-LEARNING FRAMEWORK FOR SEMANTIC-WEB," vol. 12, no. 14, pp. 2994–3006, 2021, [Online]. Available: https://turcomat.org/index.php/turkbilmat/issue/view/49.

[2] A. Kumar and N. J. Ahuja, "An adaptive framework of learner model using learner characteristics for intelligent tutoring systems," in Advances in Intelligent Systems and Computing, 2020, vol. 989, pp. 425–433, doi: 10.1007/978-981-13-8618-3_45.

[3] "Artificial Intelligence in Education (AIEd) a high-level academic.pdf," AI Ethics, Springer, vol. 2022, no. 2, pp. 157–165, 2021, doi: https://doi.org/10.1007/s43681-021-00074-z.

[4] F. Ç. Baz, "New Trends in e-Learning," in Trends in E-learning, M. Sinecen, Ed. Rijeka: IntechOpen, 2018.

[5] D. Tomar and P. Tomar, "Artificial Intelligence-Based Knowledge Representation and Reasoning," in Impact of AI Technologies on Teaching, Learning, and Research in Higher Education, S. Verma and P. Tomar, Eds. IGI Global, 2021, pp. 134–149.

[6] T. Gruber, "Ontology," in Encyclopedia of Database Systems, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, 2009, pp. 1963–1965.

[7] E. Katis, H. Kondylakis, G. Agathangelos, and K. Vassilakis, Developing an ontology for curriculum and syllabus, vol. 11155 LNCS, no. 1. Springer International Publishing, 2018.

[8] H. El-Ghalayini, "E-course ontology for developing E-learning courses," in 2011 Developments in E-systems Engineering, 2011, pp. 245–249.

[9] P. Sharma and M. Harkishan, "Designing an intelligent tutoring system for computer programing in the Pacific," Educ. Inf. Technol., no. 0123456789, 2022, doi: 10.1007/s10639-021-10882-9.

[10] A. Khan, M. Junaid, and M. Noman, "Smart Tutor an Intelligent Tutoring System for C Sharp Programming Bahria University Karachi Campus," Int. J. Comput. Appl., vol. 180, no. 27, pp. 28–33, 2018, doi: 10.5120/ijca2018916646.

[11] R. Pavlov, "Smart Tutors : Delivering Personalized Learning," pp. 1–4, 2021, [Online]. Available: https://www.epam.com/insights/blogs/smart-tutors-delivering-personalized-learning.

[12] T. Crow, A. Luxton-Reilly, and B. Wuensche, "Intelligent Tutoring Systems for Programming Education: A Systematic Review," ACM Int. Conf. Proceeding Ser., pp. 53–62, 2018, doi: 10.1145/3160489.3160492.

[13] L. Chen, P. Chen, and Z. Lin, "Artificial Intelligence in Education: A Review," IEEE Access, vol. 8, pp. 75264–75278, 2020, doi: DOI:10.1109/ACCESS.2020.2988510.

[14] N. Mendjoge;, A. R. Joshi;, and M. Narvekar, "Review of knowledge representation techniques for Intelligent Tutoring System," in 2016 International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 2508–2512.

[15] M. Masood and N. A. M. Mokmin, "Case-based reasoning intelligent tutoring system: An application of big data and IoT," ACM Int. Conf. Proceeding Ser., vol. Part F132530, pp. 28–32, 2017, doi: 10.1145/3152723.3152735.

[16] W. Abu-Dawwas and M. Abu-Dawas, "Proposed frame-based expert system to construct student's knowledge model in intelligent tutoring systems," J. Math. Comput. Sci., vol. 10, no. 5, pp. 1529–1537, 2020, doi: 10.28919/jmcs/4567.

[17] A. Tato, R. Nkambou, J. Brisson, C. Kenfack, S. Robert, and P. Kissok, "A Bayesian network for the cognitive diagnosis of deductive reasoning," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9891 LNCS, pp. 627–631, 2016, doi: 10.1007/978-3-319-45153-4_78.

[18] M. Khfagy, Y. Abdelsatar, and O. Reyad, "Knowledge Representation in Intelligent Tutoring System," vol. 533, no. 1, 2017, doi: 10.1007/978-3-319-48308-5.

[19] J. Alshboul, H. A. A. Ghanim, and E. Baksa-Varga, "Semantic Modeling For Learning Materials In E-Tutor Systems," J. Softw. Eng. Intell. Syst., vol. 6, no. 2, pp. 17–24, 2021, [Online]. Available: https://www.jseis.org/volume/vol6no2.html.

[20] A. Verma and A. Verma, "An Ontology of Ontological Engineering," Int. J. Eng. Sci., vol. 24, no. 63019, pp. 97–107, 2017.

[21] C. Ma and B. Molnár, "Use of Ontology Learning in Information System Integration: A Literature Survey," Communications in Computer and Information Science, vol. 1178 CCIS. pp. 342–353, 2020, doi: 10.1007/978-981-15-3380-8_30.

[22] S. Chimalakonda and K. V. Nori, "An ontology based modeling framework for design of educational technologies," Smart Learn. Environ., vol. 7, no. 1, 2020, doi: 10.1186/s40561-020-00135-6.

[23] T. Guber, "A translational approach to portable ontologies," Knowl. Acquis., vol. 5, no. 2, pp. 199–229, 1993.

[24] A. Ramírez-Noriega et al., "Towards the Automatic Construction of an Intelligent Tutoring System: Domain Module," Adv. Intell. Syst. Comput., vol. 930, no. 3, pp. 293–302, 2019, doi: 10.1007/978-3-030-16181-1_28.

[25] W. Yathongchai, J. Angskun, and C. C. Fung, "An Ontology Model for Developing a SQL Personalized Intelligent Tutoring System," Naresuan Univ. J. Sci. Technol., vol. 25, no. 4, pp. 88–96, 2017.