IEEE RCD STANDARD BASED ONTOLOGICAL MODELING OF COMPUTER SCIENCE CURRICULUM

Milinko Mandić[1], Milan Segedinac[2], Goran Savić[2], Zora Konjović[2]
[1]University of Novi Sad, Faculty od Education, Sombor, Serbia
[2]University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia

Abstract - *The paper presents modeling of Computer Science curriculum based on ontologies. The main concept of the proposed ontological model is the competence defined by IEEE RCD standard. Mapping of the IEEE RCD data model to ontological model is described in detail. ACM Computing classification is used for domain knowledge modeling. Protégé tool is used for designing ontologies*

1. INTRODUCTION

Specific position of Computer Science (CS) within an educational system as a whole, and dynamics of changes in the field, especially those related to an extremely frequent appearance of a new competencies, call for almost permanent adaptation of CS curricula. Therefore it is necessary to represent CS curricula in such a form that their maintenance, change and improvement can be done in a simple and quick way. In addition, CS curricula should be compliant at various levels of education [1, 2]. For example, higher education curricula for computer science teachers should be designed in a way that the graduate computer science teachers' competencies meet the needs of the current elementary and secondary school curricula. Similarly, it is necessary for students' computer education outcomes in elementary and secondary schools to meet demands of a subsequent education levels, as well as to provide students with skills and knowledge that satisfy employers' requirements for specific work posts. Moreover, it is necessary to provide a concept of curricula representation enabling an active participation in curricula maintenance and advancement for academic community, ministries in charge, students, and present employees within CS sector.

This paper presents an ontology based computer model for representing CS curriculum aimed at satisfying the abovementioned requirements.

Ontologies represent a common, multiple and repeatedly used outlook of a domain and give meaning to information structures that are exchanged by information systems. Ontology is defined, according to Gruber [3], as a formal and explicit specification of a shared conceptualization of a domain of interest, employed to help people and programs to share knowledge. "Formal" refers to the principle that ontology should be "machine readable". Ontologies are typically used in maintaining taxonomy of a given domain and modeling domain conceptual structure. Ontologies can be viewed as a semantic model that contains concepts, their properties, conceptual inter-relations and axioms related to previous elements. Ontologies may describe educational domain from various perspectives and provide enriched description and restoring of educational content [4].

Due to its ability to represent curriculum in a machine understandable manner, and the features of reuse and share, ontological approach becomes popular for representing some of the curriculum forms [5, 6]. In [5] authors state that the advantage of applying ontology to curriculum development based on the expected learning outcomes, students' competencies and standards lies in the curriculum that "forms a multidimensional matrix of information elements linked in the form of more to more links" where, according to them, ontologies are the most adequate concept. Furthermore, ontologies facilitate a change of any of the curriculum aspects regarding new recommendations obtained from accreditation bodies, standardization agencies, authorities, etc. Ontology implementation enables simple definition of the changed parts and concepts which the changes have influenced. Applying ontology supports the crucial principles that have to be met when developing a curriculum [5]:

- Dynamics of the system,
- Compatibility with the changes,
- Facility of administering, and
- Low price of maintaining.

The same source summarizes that ontology based curricula provide the following: defining firstly only the curriculum principal parts (the core) in such a way that the elementary „material“ is distinct from the latter, more advanced concepts; mapping the external criteria and the expected outcomes in the way that the developed curriculum is satisfactory; designing the curriculum content in a more systematic, rigorous and transparent mode where a larger group of teachers can participate in creating it; efficiently following and managing changes in any part of curriculum and the influence they have over the curriculum as a whole; students' personalized learning – (students are able to manage their own learning process); planning and organizing teaching activity „more precisely and adequately„; searching through the curriculum in compliance with the route adapted to an administrator, tutor or a student.

2. STANDARDS FOR REPRESENTING COMPETENCY

Competencies are defined as phase concept and represent a „useful term bridging the gap between education and job requirements“[7]. Competencies are also defined as a combination of knowledge, skills and attitude adapted to the specific situation. O*NET represents a data base of all the professions in US economy in the form of competence taxonomy where professions are represented by knowledge, skills and „various other competencies“ (known as KSAO) [8]. Students' outcomes are defined as statements of what students know, what they understand and what they are capable of doing after completion of the learning process; they actually represent a combination of knowledge, skills and competencies.

Ontological model of students' competencies within computer science curriculum in proposed in this paper is based on IEEE RCD (IEEE Standard for Learning Technology—Data Model for Reusable Competency Definitions**)** [9] standard that defines a data model for describing, referencing and sharing definition of competency, primarily for the needs of online and distributed learning. The standard provides a method for formal representation of competency key characteristics, independent of the specific usage context. Furthermore, it provides interoperability between the learning systems that contain information about the competency. This standard regards the word competency in „a very wide meaning including skills, knowledge, tasks and students' outcomes" and can be used for describing information of any of the stated notions within „the context of education, learning or training“. IEEE RCD is a widely distributed International standard (see, for example, http://www.icoper.org/ where Icoper eContentPlus system adopts the application profile of the standard). It is also a syntactic standard which makes it possible for various data systems to interoperate in the way that each system may define which part of the received data is a competency and which part of the competency refers to title, description and similar [8]. The standard stems from IMS Reusable Definitions for Competencies and Educational Objectives (RDCEO) [10]. Figure 1 represents the IEEE RCD data model standard with its components.

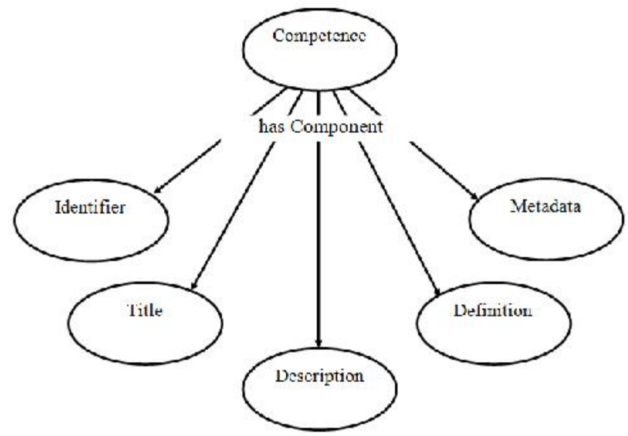

Figure 1. IEEE RCD Data Model

RCD, therefore, consists of the following mandatory components:

- a unique identifier,
- title

Optionally, it can also contain:

- description
- definition (reference to other definition or repository)
- metadata, the field which refers to further information about the specific competency (it can be of various formats, for example IEEE LOM).

2.1. Reusable Competency Definition

This subsection describes RCD data elements, and discusses advantages and disadvantages of RCD and two other standards for representing competencies. Listing 1 presents a synopsis of Reusable Competency Definition with the notation used in [9].

```
reusable_competency_definition :
record
(
identifier :
long_identifier_type,
        // Mandatory
        // Occurs 1 time
title :
bag of langstring_type(1000),
        // Mandatory
        // Occurs 1 time
        // SPM: 20 instances of langstring_type in the
        bag
        // The parameter value is the SPM for the number
        // of characters in the string element of the
        // langstring_type
description :
bag of langstring_type(4000),
        // Optional
        // Occurs 0 or 1 times
        // SPM: 20 instances of langstring_type in the
        bag
        // The parameter value is the SPM for the number
        // of characters in the string element of the
        // langstring_type
definition :
definition_type,
        // Optional
        // Occurs 0 or more times
        // SPM: 10 instances of definition in a
```

```
        // reusable_competency_definition record
metadata :
metadata_type,
        // Optional (implied default values - see 6.2.5)
        // Occurs 0 or 1 times
)
```

Listing 1. Synopsis of Reusable Competency Definition

Although the basic notions of the standard (such as title, description or definition) are represented in a human readable form as unstructured data, it is possible to provide the semantics of the standard so as for it to become understandable to computers. This can be done by adding special „knowledge“ in the metadata field and establishing relations to other RCDs by using these data in the metadata part of the standard. As the metadata field can be defined by IEEE LOM standard, when adding up semantics to RCDs it is possible to deploy IEEE LOM categories *Relation* and *Classification* (particularly *taxon* fields). Moreover, semantics might be ensured by externally relating RCDs into ontologies, which has been done in this paper.

A disadvantage of applying competency representation by RCD standard is its only partial competency representation, i.e. the standard does not explicitly include specific fields of evidence that, for example, the competencies have been attained, evidences of the mode of grading, recording, certifying competencies, etc [9]. Still, if the need arises, it is possible to extend the model in such a way that these aspects also become made explicit as RCD partially enables defining the named fields within a structured optional field of *definition* that contains zero or more model source fields and one or more elements *statement*. These can be used to define specific fields, for example *proficiency, criteria, indicators.*

This is one of the reasons why competency representation does not use IEEE draft standard “Simple Reusable Competency Mappings“ (SRCM) [11] which in addition contains a direct acyclic graph containing competencies, therefore every node has several related features such are the following [8]:

- Competency (RCD or other SCRM);
- Proficiency result (required or desirable); *Links to other nodes in the graph (parents* or children).

One disadvantage of the IEEE SCRM standard is that it currently is a draft standard. Furthermore, logical links within SCRM are not based on formal logic, and some authors [12] consider semantics defined in this model as „confusing“.

HR XML [13] standard defines competency as “A specific, identifiable, definable, and measurable knowledge, skill, ability and/or other deployment related characteristic (e.g., attitude, behavior, physical ability) which a human resource may possess and which is necessary for, or material to, the performance of an activity within a specific business context”, but foresees applying competency only within the business context.

3. AN ONTOLOGICAL MODEL OF CS CURRICULUM

This paper, therefore, proposes ontologies which are based on IEEE RCD standard representing students’ competencies domain contained within the CS curricula, so that their meaning has been formalized and computer interpretable. OWL DL language is used in practical implementation of ontology. Protégé tool (http://protege.stanford.edu) is used for designing ontologies as it is a widely and easy-to-use open source software framework for managing ontologies.

The goal of designing such a model was to provide the following:

- Availability of curriculum via Internet and/or specific software tools;
- Simple curriculum maintaining and changing according to changes in the field;
- Comparing and adapting curricula designed by various authors and institutions.

Ontological representation of competency based upon IEEE RCD standard is shown in Figure 2. The model itself is similar to the part of the model described in [12] which is related to RCD competencies. Rectangles denote ontology *classes*; lines are *object properties*, while *data type* properties are represent within the rectangular.

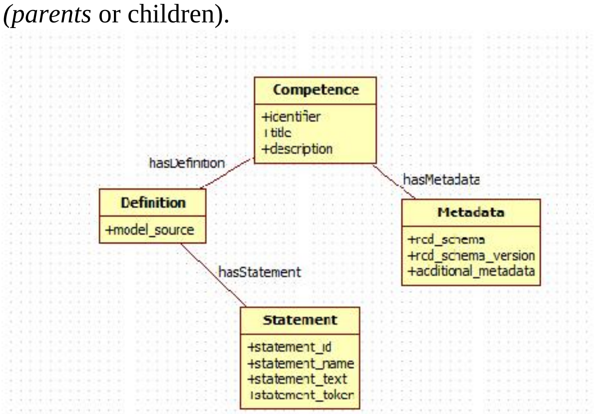

Figure 2. Ontological representation of the IEEE RCD standard

The basic class represents a competence with *identifier*, *title*, and *description* features. Since the standard defines these fields as non-structured ones, in an ontological model they describe a class in terms of data type properties. Separate classes are *Statement*, *Metadata* and *Definition*. The definition field, although being optional, represents a structured part of a competence and might contain, if necessary, one or more *statement*s, like grade, criterion, outcome, etc. It is, therefore, necessary to represent it with a specific class related to the *Statement* class via *hasStatement* object property. Similarly, metadata field is described by a separate class *Metadata* (IEEE RCD standard states that this field can also be presented using IEEE LOM standard). Cardinalities of features in ontological model correspond to the underlying standard; as in IEEE RCD standard a competence has exactly one identifier and title, and zero or one metadata field, object and data type properties are of cardinality as presented in Figure 3.

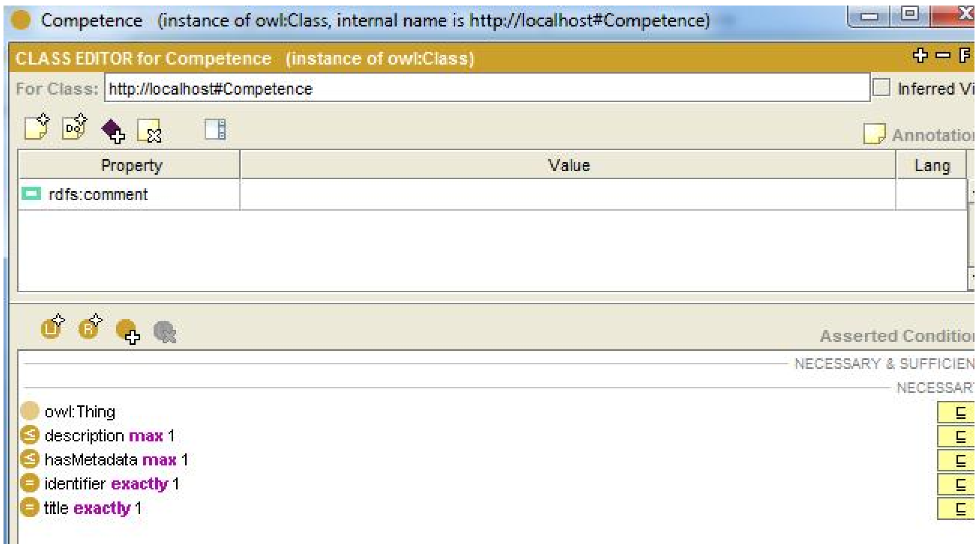

Figure 3. Cardinalities of object and data type properties for the *Competence* class

In our model upper ontology classes are:

- *Competence,*
- *EducationalProgram,*
- *Statement,*
- *Metadata*, and
- *Definition.*

Competence class is represented according to the IEEE RCD standard as shown in Figure 2. As the paper is focused to ontological representation of a knowledge and gained skills, with having in mind the possibility of a wide interpretation of competencies in IEEE RCD standard, competencies are here defined as knowledge and skills. Therefore the *Competence* class is parental to classes *Knowledge* and *Skills*. *Knowledge* class is equivalent to curriculum thematic fields, while *Skills* represent pupils'/students' expected gained skills after the specific education program has been completed. The specific outcome (*Skills*) of education program can be related to one or more curriculum thematic fields via object property *hasKnowledge*. *EducationalProgram* (with *identifier*, *title*, *educational_level* data type properties) is related to the *Competence* class via *hasCompetence* object property.

There is a number of proposals related to content and structure in CS curricula with the continual trend of enlarging the number of them. It is clear that there are several sources influencing CS curricula design: leading professional and academic associations (IEEE, ACM), Ministries in charge, accreditation bodies, educational institutions themselves, employers, etc. In this paper we have rest upon the content of CS curriculum as specified in ACM recommendation for CS domain [14], while thematic fields (*Knowledge*) have been modelled in accordance with ACM Computing classification [15]. The earlier version of ACM computing classification has been ontologically represented as a taxonomy which is available at http://daml.umbc.edu/ontologies/classification. The taxonomy categorizes the following subclasses within the *Knowledge* class:

- General_and_reference
- Hardware
- Computer_Systems_Organization
- Networks
- Software_and_its_engineering
- Theory_Of_Computation
- Mathematics_Of_Computing
- Information_Systems
- Security_and_privacy
- Human_centered_computing
- Computing_Methodologies

- Applied_computing
- Social_and_professional_topics

Each of the listed topics is further divided into its subtopics that represent its subclasses which, again, may have their own subtopics related to their subclasses, etc. For example, the direct subclasses of the *Hardware* class are the following ones:

- Printed_circuit_boards
- Communication_hardware_interfaces_and_storage
- Integrated_circuits
- Very_large_scale_integration_design
- Power_and_energy
- Electronic_design_automation
- Hardware_validation
- Hardware_test
- Robustness
- Emerging_technologies

Class *Metadata* does not have subclasses defined in our model because, for now, the model is not intended to include learning objects. But this class allows for easy extending the model with learning objects (in accordance with, for example, the IEEE LOM standard) which, in turn, can be simply related to the thematic areas (*Knowledge*) through appropriate object properties. Also, so far subclasses of classes *Definition* and *Statemen*t are not further defined, since they are optional fields, and defined model does not imply the existence of concepts such as criteria, methods for assessment, and the like, but if the need arises ontology can be easily to extended by applying appropriate subclasses.
Listing 2 presents a part of the owl code of the proposed ontological model .

```
<owl:Class rdf:about="#Educational_program">
    <rdfs:subClassOf
rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
        <owl:onProperty>
          <owl:DatatypeProperty
rdf:ID="educational_level"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
        <owl:onProperty>
          <owl:FunctionalProperty
rdf:about="#identifier"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:FunctionalProperty rdf:about="#title"/>
        </owl:onProperty>
        <owl:cardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#Statement"/>
    <owl:disjointWith rdf:resource="#Definition"/>
    <owl:disjointWith rdf:resource="#Competence"/>
    <owl:disjointWith rdf:resource="#Metadata"/>
  </owl:Class>
<owl:Class rdf:about="#Skills">
    <owl:disjointWith rdf:resource="#Knowledge"/>
    <rdfs:subClassOf rdf:resource="#Competence"/>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom rdf:resource="#Knowledge"/>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="hasKnowledge"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
```

Listing 2. A part of CS curriculum ontology

4. CONCLUSION

This paper proposes ontological modeling of a CS curriculum. The main concept of the proposed ontological model is the competence defined by IEEE RCD standard. Curriculum content and structure are modeled following the ACM CS standard and the ACM Computing taxonomy.

Although the proposed model does not contain classes explicitly describing aspects referring the mode of grading, certifying, evidencing competencies attained, etc., these aspects are possible to define without extending the model by the use of *Statement* and *Definition* classes of the proposed model.

Even with such refinements included, the presented model does not describe students' skills well enough, so one of the future work directions is to improve the model in order to provide mechanism for sufficient representation of students' skills. The second, more practical research direction foreseen by authors is to create a graphical user interface that would facilitate curriculum manipulation for those users who are not familiar with general ontology design and maintenance tools.

5. REFERENCES

[1] A. Tucker, "*A Model Curriculum for K-12 Computer Science*", The ACM K-12 Education Task Force, 2004, http://csta.acm.org/Curriculum/sub/CurrFiles/K-12ModelCurr2ndEd.pdf, Retrieved January, 2013.

[2] M. Mandić, Z. Konjović and P. Viđikant, "*The profile of high school informatics teachers in the autonomous province of Vojvodina*", Croatian Journal of Education, 2013, in print.

[3] T. Gruber, "*Toward principles for the design of ontologies used for knowledge sharing*", International Journal of Human and Computer studies, 1995, *43*(5/6), 907-928.

[4] A. Elsayed, "*Interaction with Content through the Curriculum Lifecycle*", Advanced Learning Technologies, 2009. ICALT 2009, pp. 730 – 731

[5] H. Dexter and I. Davies, "*An ontology-based curriculum knowledgebase for managing complexity and change*", Ninth IEEE International Conference on Advanced Learning Technologies, 2009, icalt, pp.136-140.

[6] J.T. Fernández-Breis, D. Castellanos-Nieves, J. Hernández-Franco, C. Soler-Segovia, M. del Carmen Robles-Redondo, R. González-Martínez and M. P. Prendes-Espinosa, „*A semantic platform for the management of the educative curriculum*", Expert Systems with Applications, 2012. 39(5): pp. 6011-6019.

[7] J. Boon and M. R. Van der Klink, "*Competencies: the triumph of a fuzzy Concept*", International Journal of Human Resources Development and Management, 2003, 3:pp.125–37.

[8] K. Lundqvist, K. Baker, S. Williams, „*Ontology Supported Competency System*", International Journal of Knowledge and Learning, Volume 7, Numbers 3-4, December 2011 , pp. 197-219(23).

[9] IEEE Standard for Learning Technology—Data Model for Reusable Competency Definitions, http://www.cen-ltso.net/main.aspx?put=264

[10] IMS Reusable Definition of Competency or Educational Objective Specification, http://www.imsglobal.org/competencies/, Retrieved January, 2013.

[11] Draft Standard for Learning Technology— Simple Reusable Competency Map, http://www.cen-ltso.net/main.aspx?put=1054

[12] J. De Coi, E. Herder, A. Koesling, C. Lofi, D. Olmedilla, O. Papatreou, W. Siberski, "A Model for Competence Gap Analysis", Proceedings of 3rd International Conference on Web Information Systems and Technologies (WEBIST), 2007, Barcelona, Spain.

[13] HR XML, http://www.hr-xml.org/?page=DownloadLandingPage, Retrieved January, 2013.

[14] ACM Computer Science Curriculum, http://www.acm.org/education/curricula/ComputerScience2008.pdf, Retrieved January, 2013.

[15] The 2012 ACM Computing Classification System, http://www.acm.org/about/class/2012, Retrieved January, 2013.

ACKNOWLEDGEMENT

The research presented in this paper is partially funded by the Serbian Ministry of Education, Science and Technology Development, Grant No. 47003.