Je-LKS
Journal of e-Learning and Knowledge Society
The Italian e-Learning Association Journal
PEER REVIEWED PAPERS
PERFORMING ARTS, MEDIA AND TRAINING
Vol. 12, n.1, 2016
ISSN: 1826-6223 | eISSN: 1971-8829

ALGOSKILLS: AN ONTOLOGY OF ALGORITHMIC SKILLS FOR EXERCISES DESCRIPTION AND ORGANIZATION

Takie Eddine Belhaoues[1]
Tahar Bensebaa[1]
Meriem Abdessemed[1]
Anis Bey[1,2]

[1] Computer Science Department, Laboratory of Research in Computer Science (LRI), Badji Mokhtar-Annaba University, Annaba, Algeria
[2] Ecole Préparatoire en Sciences Economiques, Annaba, Algeria

Keywords: Ontology, Skills, Algorithm, Learning resources, Semantic organization.

Algorithms and data structure is considered as an important course in computer science where students have to learn several fundamental principles of programming. Where many tools have been developed to aid learning programming this work tries to represent the domain knowledge of programming that can be used easily and efficiently by these tools. AlgoSkills is an ontology of algorithmic skills that represents this domain knowledge and developed to be used in any pedagogical task and essentially in the organization of learning resources. A first experience has been conducted to use this ontology to build a basis of algorithmic exercises semantically and pedagogically structured. This basis is initially exploited by Algo+, an assessment tool of algorithmic competencies.

for citations:
Belhaoues T.E., Bensebaa T., Abdessemed M., Bey A. (2016), *AlgoSkills: An ontology of Algorithmic Skills for exercises description and organization,* Journal of e-Learning and Knowledge Society, v.12, n.1, 77-92. ISSN: 1826-6223, e-ISSN:1971-8829

1 Introduction

Algorithms promotes, among other things, working and thinking methods acquisition, analytical skills development, abstraction, anticipation and logic (Müldner & Shakshuki, 2003). This is a field of "know-how". Algorithms teaching consists in passing the algorithm designing rules and techniques; in other words, in describing the required problems solving steps. However, the teaching and the learning of this field are particularly complex due to its abstracted knowledge requiring from teachers considerable efforts to find adequate methods assimilating abstract concepts to novice students. These students can meet various constraints during their learning, which could alter their reasoning ability.

By taking into account these considerations, we support that the best way to learn and assimilate the algorithmic concepts and skills is practicing on exercises treating the good pedagogical objectives. At the University of Badji Mokhtar, Algeria, the overarching goal of Computer Science Department is to increase students' proficiency in programming. The first experience is the use of the automatic assessment tool Algo+ (Bey & Bensebaa, 2011). The latter includes skills such as algorithmic problem solving, abstraction and top-down design.

However, the exercises set handled in Algo+ are devoid of meanings and characteristics between them. This lack represents a brake to the enormous potential of exploiting these educational resources and their interconnections. To that end, an organization of exercises collection based on ontology specificities may represent an interesting solution.

There are several definitions of ontologies in the literature, the most common (Gruber, 1995): *An ontology is an explicit specification of a conceptualization*. Since the 90s, ontologies have become a research subject at the heart of different communities, including artificial intelligence, semantic web, software engineering, biomedical informatics, or the information architecture, etc. The reason for this popularity is partly due to the fact that ontology is a controlled and organized vocabulary and corresponds to the explicit formalization of the relations created between the various vocabulary terms. On the other hand it offers a common and shared understanding of a domain, as well as human users and at the level of software applications. According to Desmoulin (Desmoulins & Granbastien, 2006), ontological engineering provides to TEL "Technology-Enhanced Learning", a structured and explicit representation, shared by a community (human or artificial agents), which makes reference and guides their conception. It can also supply a formal representation, coupled with an inference mechanism allowing, among others, to these environments to do "reasoning" (Psyché, Mendes & Bourdeau, 2003).

Taking into consideration the nature of the field that focuses on how to solve problems by manipulating a set of knowledge and the specificities offered by ontologies in terms of representation and formalization coupled with a reasoning mechanism, we propose in this work an algorithmic ontology grouping the knowledge by the theoretical notions of the domain, the know-how by the skills required to solve algorithmic problems, and the semantic relationships that may arise between the concepts of ontology. The resulted ontology will represent a solid and appropriate foundation to respond to two necessary and complementary needs as illustrated in figure1.

The first need is to build a base of exercises semantically and pedagogically organized. To do so, the domain's concepts (notions and skills) and the relations between them are exploited in formal and explicit structure in order to enhance the description of each exercise put available to students through Algo+. The use of ontologies offers by their specificities this semantic contribution lacking in the previous formalisms. The pedagogical aspect is justified by the nature of the concepts and relations of this ontology, which formalizes a set of skills and notions of the algorithmic field.

As second need, appropriated algorithmic exercise has to be proposed to each learner situation. To achieve this, we plan to take advantage of the explanation, representation and formalization of knowledge (concepts and skills) expressed through the ontology of algorithmic. Therefore, we have to accompany the learner skills acquisition in order to bring him to better problem solving by proposing adapted training path to his profile. For that purpose, construction and use of the ontology were oriented to a logical sequence of the proposed exercises. This part of work is not covered in this article.

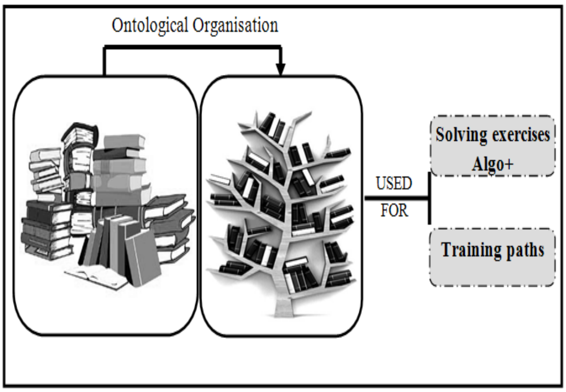

Fig. 1 - Ontological organization of exercises and its uses

A study of the skills ontologies allowed us to notice that no ontology of algorithmic notions and skills was previously built. However, two works including ontologies skills have interested us.

This absence of explicitation and formalization of the domain knowledge incited us to build an ontology of algorithmic skills. This paper concerns essentially the presentation of the ontology **AlgoSkills** which constitutes the core of our system. A tool exploiting the ontology is available to users-teachers in order to provide the algorithmic exercises base of semantic and pedagogical organization. This base is then used by the Algo+ skills assessment tool.

This paper is structured as follow, section 2 present some related works. In section 3, the aim and the methodology of ontology construction are detailed. Section 4 introduces the realized indexing tool, and its uses. Finally, section 5 concludes our article and gives some orientations for future works.

2 Related works

2.1 Ontologies of skills

To the authors' knowledge, there are not ontologies skills in the field of algorithms; however, we could identify only two ontologies representing skills treating other domains.

The first one, used in an European project is "The Ontology Skills of dynamic geometry" called **GeoSkills** (Desmoulins, 2010); it contains classes and instances representing competencies from the mathematics education standards of Spain, Germany, France and United Kingdom. This ontology is the core element of the Intergeo project, aims at providing teachers with means to share dynamic geometry resources across Europe. It has developed a platform[1] where new resources can be added and search for existing ones by subject, level, and instructional type executed (Desmoulins & Libbrecht, 2009). GeoSkills essential ingredients are topics, competencies, pathways, levels and programs.

The second ontology is named **CogSkillNet** "Ontology-Based Representation of Cognitive Skills": it represents the cognitive skills for K-12 education domain (primary and secondary education) and describes the development and design processes of skill ontology. CogSkillNet is one of the two ontologies - the other one is concept ontology (ConceptNet) - embedded in the POLE e-learning system. CogSkillNet models the cognitive skills, which are associated with two repositories (learning object repository and assessment object repository) (Askar & Altun, 2009).

1 http://i2geo.net

2.2 Exercises Databases in Algorithms

We found in the literature mainly tow databases of algorithmic execises.

EDBA Project (Exercises DataBase about Algorithms)[2] is a project of the research laboratory (LIG) at the University of Grenoble I. J.Fourier, having for objective in the field of computer education, to set up a base of algorithmic exercises, associated with a programming environment to the educational community -teacher and students-. The development of EDBA project comprises three successive web versions: EDBA0.0, EDBA 1.0, EDBA 2.0. The first application EDBA 0.0 consists of a mini-programming environment allowing to asses a program with sets of recorded tests. The last ones are extended by the construction of exercises database, the introduction of both classes of users (teachers and students) and the participative aspect of the latter. In EDBA, the exercises are structured in a database. They are identified by the name of the function or program. They are characterized by a statement in natural language, a semi-formal specification of the header of the function and test sets. Example exercise: search for an item in a sorted list. EDBA 1.0 adds a new feature to the exercises, which is their ranking on a scale of difficulty with several levels.

AlgoBank2010 (Algorithmic Database)[3] is a Collaborative project of the UTC (University of Technology Compiegne), whose objective is to capitalize exercises in the algorithmic field by creating a pedagogical resource center to introduce programming for computer beginners. The pedagogical resource center contains several exercises capitalized in the same database, statements, aid and corrections as well as examples of programs associated, in programming languages (Pascal, Java, C) in PDF format. The database is available for teachers and students. It is therefore only topics of exercises with their answers not complete courses or simulations or interactive exercises. They are limited to simple documents, of one or two pages generally, as PDF files. A demonstrator achieved within a first project Algobank2008 is integrated into the moodle[4] platform.

We note that in both environments, the exercises are stored in databases, a structure allowing them exclusively to be easily accessible by the users. Besides, this organization is considerably lacking of semantic, and this is what is observed in most algorithmic resources (particularly the tests and exercises). Indeed, Bouhinau references some works and mentions in (Bouhinau, 2010) that in computer science, in particular for teaching algorithms, However re-

[2] http://metah.imag.fr/?page_id=78

[3] http://www.unit.eu/fr/membre/projets_unit/2010/228

[4] http://moodle.utc.fr/course/view.php?id=503

sources are available, but there is huge lack of structuration or pedagogical, semantic and epistemological organization. As we presented in the first section, one of our objectives is actually to remedy this deficiency by proposing an exercise base with semantic and pedagogical organization based on ontology of skills. This exercise base can be used by Algo+ as by other environments of algorithmic evaluation or learning.

3 AlgoSkills; an ontology of algorithmic skills

In this section, are presented the steps followed in the ontology-design process of AlgoSkills, the different components (class, relation..) and their characteristics. But first of all, what is meant by "skills"?

3.1 Skills in Algorithms

The most classic definition divides the skills in knowledge, know-how and know-be (Katz, 1974). Guy Le Boterf proposes in his part that: "*Skill is the mobilization or activation of several types of knowledge in a situation and context*" (Le Boterf, 1995). The definition that seems the most appropriate to our context is the one of the High Council of French education[5], which declares that: *"A skill is always a knowledge combination, capacities to implement this knowledge, and of attitudes that is states of mind necessary for this implementation"*. For example the skill: *"Choose between use loop For and loop While"*, require knowledge about the notions loop *For* and *While*, as capacity being able to distinguish and choose which loop should be used depending on the context. Therefore, it is not obvious to give a definition of "skill", and this is not without consequences for its use. Difference between types of skills is also noticed.

In the development of ontology, we should start by defining the domain covered, and identify who will use and maintain the ontology (Noy & McGuinness, 2001). The proposed ontology is comparable to GeoSkills (Desmoulins & Libbrecht, 2009) in sense that both of them are representing an educational field.

Given that algorithms is a discipline which constitutes the resolution of problems. The ontology based representation can offer a good organization of knowledge. Particularly the by the skills set, which is fundamental in a know-how field. It consists to specify the basic concepts, outsource and formalize the skills, for then explain the relationships between them. The ontology AlgoSkills includes concepts that describe the basic notions taught in algorithmic courses, and skills required to solve algorithmic problems. It includes also, among others, erroneous student skills often committed during their learning

[5] http://pedagopsy.eu/definition_des_competences.html

(e.g. *infinite loops*), as well as exercises and their solutions as resources. The goal is to organize an algorithmic exercises base on these skills. This means that, each exercise statement treat one or several skills for such Topics. The current user of the ontology is the teacher of the domain. It has for leading role to describe algorithmic exercises with the adequate concepts (notions and \ or skills treated), possibly making updates in the base which is gradually built, enrich the ontology with new skills, etc.

3.2 Ontology-design process

After having defined the domain and the ontology objectives, we present in the following the construction methodology. Several methodologies assisting ontologists in the different construction step have been developed (Fernandez, Gomez-Perez & Juristo, 1997; Bachimont, Isaac & Troncy, 2002), however there is no methodology making the unanimity. We choose the construction method proposed by Noy and McGuinness (2001), which fit our working context; it includes the various necessary steps for the ontology design. It is both, simple, detailed and explained; and whose steps treat gradually each ontology component:

1. Enumerating Important Terms
2. Defining the classes and Class Hierarchy
3. Defining properties and Relations
4. Creating Instances

The ontology is defined in OWL language. Recommended by the W3C in February 2004, it is the most expressive ontology language for the Web. It offers features that were not defined by other W3C languages, RDF (Resource Description Framework) and RDFS (RDF Schema) (Lacot, 2005; Hankel, 2000).

Protege was used as an ontology editor. PROTÉGÉ-II is a suite of tools and a methodology for building knowledge-based systems and domain-specific knowledge-acquisition tools. *Protege* (http:// protege.stanford.edu) is an open-source tool that allows developers to create and to manage terminologies and ontologies. It is more than a terminology-editing tool, as it also provides a platform for developers to use the terminologies in end-user applications. Protege is actively supported by a strong community of users and developers that field questions, write documentation, and contribute plug-ins. There are more than 70,000 registered users of Protégé who are using the system to manage terminologies and ontologies in many different domains. (Horridge *et al.*, 2004).

Enumerating Important Terms. This initial stage is useful in any ontology-design process. It capitalizes algorithmic resources on which we have worked (Donald, 1999) with expert teachers in the field, in order to extract terms considered as important. The similar terms are removed sequentially. According to McGuinness (Noy & McGuinness, 2001), it is important to get a comprehensive list of terms (array, variable, function...). Without worrying about overlap between concepts they represent, relations among the terms, or any properties that the concepts may have, or whether the concepts are classes or slots.

Defining the Classes and Class Hierarchy. The next step is the most important in the ontology-design process. It is intertwined with the properties definition. In fact, there are several possible approaches in developing a class hierarchy (Uschold & Gruninger, 1996). The Bottom-up approach was followed. Starting from the fine granularity concept (most specific) to more generic concepts. Then with an iterative process, concepts were improved and restructured if necessary. The classes are labeled in two languages (English/French), this fact is represented by using meta properties rdfs:label and by specifying an attribute xml:lang.

The ontology is organized in three super-classes: Class Topics, Class Skills and Class Resources that includes and describes the exercises and their solutions.

Class Topics: Represents the hierarchy of the basic notions usually taught in algorithmic courses. Each topic concept is characterized by name, a comment that defines it, a subclass if there is, could be disjoint with other, classes. For example, “function” is a topic and a subclass (sub topic) of “Algorithm structure”

Class Skills: Let us give a brief definition of two types of skills in education, on which the approach was based:

- **Disciplinary skill**: Know how based on the mobilization and the efficient use of specific resources set to every discipline. The development of the skill is the main aim of the discipline (Québec Education Program, 2013).
- **Transversal skill**: Know how based on the mobilization and the efficient use of common resources set to the various learning domains. Transversal skills are intellectual, methodological, personal and social. They are extremely important in the achievement of learning situations and are constantly changing (*Ibidem*).

Fig. 2 - Branch of Class hierarchy skills in AlgoSkills

We have therefore decomposed the algorithmic skills in two high categories (top level concepts), disciplinary and transversals. The disciplinary skills, as their names suggest, are strictly related to the field. We respectively categorized them in skills of *Conception, Evaluation, Problem Solving* and *Technical*. The transversals for their part are divided into skills not necessarily related to the field: *compare, justify*. The skills defined and associated with these categories, are described by a verb at the lowest level and applied to a Topic.

Example: In technical skills, which are decomposed into skills of *Data structure, Algorithm structure, Control structure*, etc. Each sub concept of the latter (Skills of the lowest level) is described by a verb.

The problem solving skills were extracted from a Course designed to aid students develop algorithmic problem-solving skills (Muller & Haberman, 2009). The following table illustrates skills and their associated supporting activities.

Table 1
PROBLEM-SOLVING SKILLS AND ASSOCIATIVE ACTIVITIES

Problem-Solving Skill	Learning/Instructional Activity
Problem's comprehension	Reformulation of the problem statement in terms of initial state, goal, assumptions and constraints.
Problem's decomposition	Identifying, naming and listing subtasks.

Problem-Solving Skill	Learning/Instructional Activity
Analogical reasoning	Identifying similarities between problems. Distinction between structural and surface similarities. Raising awareness of common mistakes caused by referencing to unsuitable problems.
Generalization and Abstraction	Extracting prototypes of problems from analogical problems in different contexts.
Identifying problem's Prototype	Systematic and well-structured introduction of algorithmic patterns. Problems' categorization.
Problem's structure Identification	Identifying the relation between subtasks. Schematizing a problem's structure, using diagrams.
Evaluation and appreciation of efficiency and elegancy	Comparing solutions with regard to efficiency and elegancy.
Reflection and drawing Conclusions	Reflecting on problem-solving processes and strategies. Making conclusions for the future.
Verbalization of ideas	Formulation of a precise idea, differentiating between an idea and its implementation.

Relations and Attribute. The "**is_a**" relation represents the hierarchy of concepts and sub-concepts; e.g.: *Loop-for* is_a sub class of *iterative statements*. Topics represent the learning theoretical knowledge of the field and are useful for the acquisition of skills. The concepts of the Topics hierarchy are related with a semantic relation "**is_useful_to**" to the appropriate disciplinary skills. E.g. the topic *Recursivity* have to be used for the disciplinary skills of conception which aim to *identify_the_recursives_structures_of_the_problem*.

The practice of algorithms is a process of a problem solving where exercises aim to concretize the good use and the manipulation of skills in order to certify their mastery. The semantic relation "**has_for_objective**" attach each exercise to at least one skill.

The exercise's statement, which formulates and explicit its data and aim, represents a Datatype Property (attribute) of the class exercises.

Creating Instances. Individuals or instances in the ontology are the concrete elements described by classes. The instantiation is the last step in the design process, but it still considered as a sensitive stage, where several points of view can exists without being erroneous nonetheless. It is however necessary to take a choice and to justify it with regard to its need. To be able to exploit the relation "**is_useful_to**" and thus specify which topics are necessary for such disciplinary skills, we have instantiated skills and topics at the lowest level, because Object Properties may be used to relate one individual to another as described in figure 3.

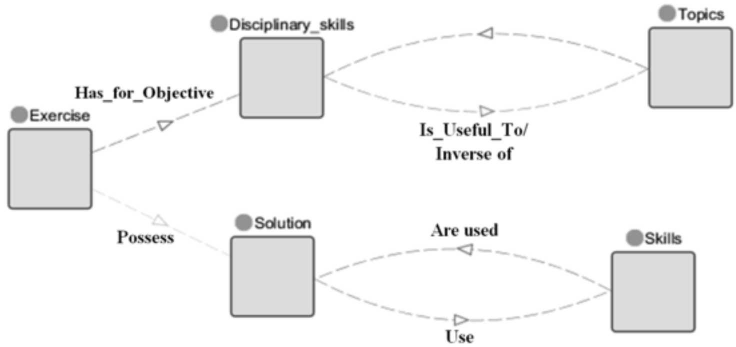

Fig. 3 - ObjectProperty View

4 The role of ontology in learning resources organization

Elicitation and formalization of topic and skills knowledge in algorithms through AlgoSkills presented in previous section constitute de pillar of our objectives defined above.

We will present in this section a developed tool allowing us to overcome this contemplative level and act with AlgoSkills ontology. The tool has to provide to *Algo+* an exercise base having *a semantic and pedagogic structure*.

Indeed, compared with the various education domain, algorithmic resources, in particular the training tests and exercises, used by learning systems are mainly dependent on a given programming language (Lisp, prolog, C) (Bouhinau, 2010). Algo+ (Bey & Bensebaa, 2013) proposes to formulate solutions in pseudo-code. We are convinced that obtaining a good program must be from a well designed algorithm. If the problem is well understood and decomposed, and the algorithm well designed, it could be translated into any existing language to obtain a functional program.

4.1 *Describing exercise and populating ontology*

In order to build a semantic and pedagogic structure of the exercise base, we constructed our work around the ontology. The latter allow teacher to describe an exercise throughout a set of elements. Populating ontology allows us to exploit the specificities of this knowledge representation formalism by instantiating concepts ontology and relation between them.

The tool is based on considering every problem to solve as an instance of ontology concept “Exercise”. Thus, describing an exercise will be guided by the manner that “Exercise” concept is elicited on the ontology. It returns therefore to use the DataProperty “statement” and the Object property “has_for_objective" with the concept "Skills" to describe the exercise in the base.

When adding a new exercise, a Java interface offers to the teacher a text area in which s/he introduces the exercise statement in natural language. This description aims to explain to learners the main objectives of the exercise.

As algorithmic field is centered on know-how knowledge and practical competencies, we have considered that each problem to solve emphasizes one or more skills; this has been taken into consideration in the ontology development under the ObjectProperty "has_for_objective". All of this strengthens our choice to include ontology skills as exercise descriptor.

The tool offers to user the possibility of navigating the ontology through hierarchical and semantic relations. The JGraphx library[6] is used in the graphical view development, which help in drawing 2D graph in java application.

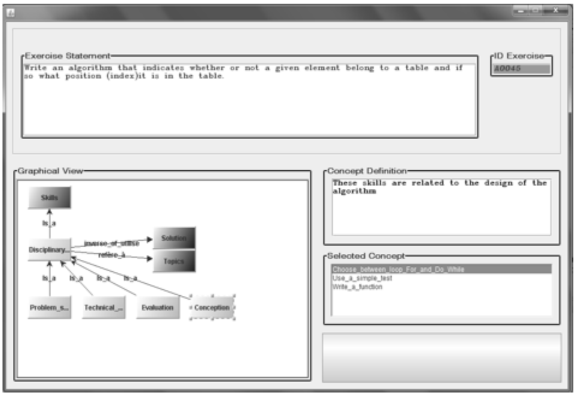

Fig.4 - Screenshot of the edit mode that enables ontology-based description of the exercise

Indeed, browsing the ontology allows users to recognize knowledge organization and structure and to identify through the concept definition extracted from annotation in ontology, those who are related to the concerned exercise. If teacher considers that one skill is an objective for the described exercise, s/he adds it to the list of its related skills. It is important to note that the skills that can be selected are situated at the lower level of the hierarchy, i.e. leaf concepts representing specific skills.

The ontology populating allows us borrowing from ontology concepts, with their attributes and semantic relation (data and object properties), associated to the descriptor exercise. Indeed, for each described exercise, an instance of the concept "Exercise" is created. Thus, "ID" and "Statement" introduced

[6] http://jgraph.github.io/mxgraph/docs/manual_javavis.html

through interface will constitute respectively the name and the DataProperty "Statement" value of the created instance.

For their part, all selected skills via the graphical view will be related to the exercise by the ObjectProperty "has_for_objective". The domain of the ObjectProperty will take as value the exercise instance, while the range will take respective instance of each selected skills.

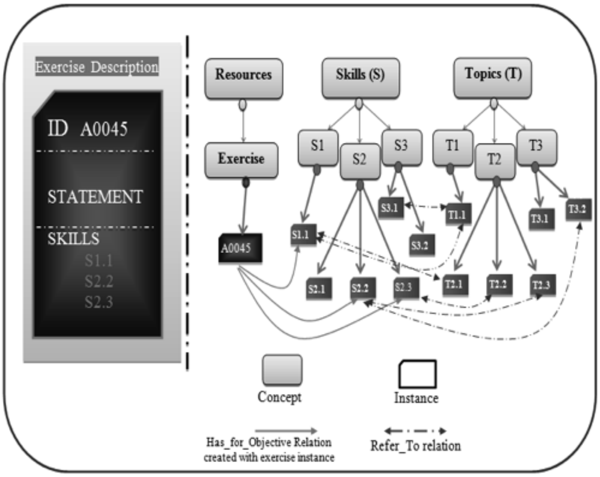

Fig. 5 - bPopulating ontology by concepts instantiation

The source code below is extracted from the owl file, after generation of the full description of an exercise with the edit mode presented previously.

```
<Exercise rdf:ID="A0045">
<Statement rdf:datatype="http://www.w3.org/2001/
XMLSchema#string"> Write an algorithm that indicates whe-
ther or not a given element belong to a table and if so what
position (index)it is in the table.</Statement>
<Has_objective>
     <Identify_the_output_datardf:ID="Identify_the_output_
     data_1"/>
</Has_objective>
<Has_objective>
     <Write_a_functionrdf:ID="Write_a_function_1"/>
</Has_objective>
<Has_objective>
     <Reuse_a_solutionrdf:ID="Reuse_a_solution_1"/>
</Has_objective>
<Has_objective>
     <Use_loop_Forrdf:ID="Use_loop_For_1"/>
</Has_objective>
<Has_objective>
     <Identify_the_input_datardf:ID="Identify_the_input_
     data_1"/>
```

```
</Has_objective>
<Has_objective>
     <Identify_variablesrdf:ID="Identify_variables_1"/>
</Has_objective>
<Has_objective>
     <Identifying_problem_prototyperdf:ID="Identifying_pro-
     blem_prototype_1"/>
</Has_objective>
<Has_objective>
     <Choose_between_loop_For_et_whilerdf:ID="Choose_betwe-
     en_loop_For_et_while_1"/>
</Has_objective>
</Exercise>
```

The browsing, access and instantiation of the ontology were done using the API Jena which offers a set of methods for manipulating ontologies from Java application (Rajagopal, 2005).

Conclusion and future works

Learning programming and algorithms remain indefinitely an essential part of the training curriculum, going beyond the computer science field. However, the learning path often gives a great deal of difficulties to the different involved actors. This is actually due to the domain concepts and skills to be acquired that are not only new for novice learners, but are also abstracted; requiring algorithmic thinking, not yet developed. This makes the role of the teacher rather sensitive

In this article, we have presented an algorithmic ontology including not only the domain knowledge, but also skills required for the domain learning. This ontology has for objective to remedy the problem faced by teachers and learners in the teaching/learning activities.

The choice of using ontology is justified by its principle.The last one defines this concept as means to explicit the domain knowledge and to formalize them, in a way to enable reusing the resulted model.

A methodology (Noy & McGuinness, 2001) has been studied and chosen to construct AlgoSkills ontology. This methodology is characterized by successive steps, ranging from why develop an ontology, and explaining gradually every detail to define its different elements.The ontology is used to propose a semantic and pedagogical organization of learning resources. Before being proposed to the learner in Algo+ system (Bey & Bensebaa, 2013), exercises were described on the basis of the knowledge and skills extracted from the ontology.

This approach opens us several axes that can be explored. Based on the semantic and pedagogical organization of the exercises, detailed in this article, we plan in future work to study the sequencing of exercises in the form of training paths, adapted to each learner using Algo+. Also, reasoning on the ontology to parameterize the rules for the training adaptation could be explored.

REFERENCES

Abel, M. H. (2004), *Utilisation de normes et standards dans le projet MEMORAe*, CAIRN, Distances et savoirs 2004/4, Volume 2, p. 487-511.

Askar P., & Altun A. (2009), *CogSkillnet: An ontology-based representation of cognitive skills*, Journal of Educational Technology & Society, 12(2), 240-253.

Amardeilh, F. (2007), *Web Sémantique et Informatique Linguistique: propositions méthodologiques et réalisation d'une plateforme logicielle*, Doctoral dissertation, Université de Nanterre-Paris X.

Bachimont B., Isaac A. & Troncy R. (2002), *Semantic commitment for designing ontologies: a proposal*, Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, Springer Berlin Heidelberg, Volume 2473, pp 114-121.

Behaz A. & Djoudi M. (2009), *Approche de Modélisation d'un Apprenant à base d'Ontologie pour un Hypermédia adaptatif Pédagogique*, In CIIA.

Bey A. & Bensebaa T. (2011), *Algo+, an assessment tool for algorithmic competencies*, Global Engineering Education Conference (EDUCON), IEEE, P 941 – 946.

Bey A. & Bensebaa T. (2013), *Assessment makes perfect: improving student's algorithmic problem solving skills using plan-based program understanding approach*, International Journal of Innovation and Learning, Volume 14 (2) – pp 162-176.

Bouhinau D. (2010), *Projet pour une base d'exercices liée à un environnement d'apprentissage pour favoriser des développements mutuels via une démarche participative*, URL: https://bdenisb.files.wordpress.com/2012/02/projet_edba_1-1-8.pdf (accessed on 2 October 2015).

Desmoulins C. (2010), *Construction avec des enseignants d'une ontologie des compétences en géométrie, Geoskills*, In 23es Journées francophones d'Ingénierie des Connaissances (p. 13).

Desmoulins C. & Granbastien M. (2006), *Une ingénierie des EIAH fondée sur des ontologies, Environnements informatiques pour l'apprentissage humain*, p. 161-179.

Desmoulins C. & Libbrecht P. (2009), *CompEd, a web-based competency ontology editor for dynamic geometry*, In Proceedings of SWEL workshop.

Donald E. K. (1999), *The art of computer programming*, Sorting and searching, 3, 426-458.

El Hassan A., & Lazrek A. (2007), *Des ontologies pour la description des ressources pédagogiques et des profils des apprenants dans l'e-learning*, 1ere Journées Francophones sur les ontologies.

Fernandez M., Gomez-Perez A. & Juristo N. (1997), *Methondology: from Ontological Art Towards Ontological Engineering*, AAA-97 Spring Symposium on Ontological

Engineering, Standford University.

Garrot É. (2006), *Un système pour conseiller le tuteur de situations d'apprentissage collaboratives*, In Rencontres Jeunes Chercheurs en EIAH (pp. p-99).

GRUBER T.R. (1995), *Toward principles for the design of ontologies used for knowledge sharing?*, International journal of human-computer studies, vol. 43, no 5, p. 907-928.

Hammache A. & Ahmed-Ouamer R. (2006), *Système d'Inférence pour une Indexation de Documents Basée sur une Ontologie de Domaine*, In INFORSID (pp. 895-910).

Hankel, J. (2000), *Resource Description Framework, More Semantic for the Web*. CSCI 2000, URL: http://www.cs.colorado.edu/~kena/classes/7818/f00/presentations/rdf.pdf (accessed on 25 September 2015).

Horridge M., Knublauch H., Rector A., Stevens R., & Wroe C. (2004), *A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools Edition 1.0*, University of Manchester.

Katz R. L. (1974), *Skills of an effective administrator*, Harvard Business Review, 1974, Vol. 51.

Lacot, X. (2005), *Introduction à OWL, un langage XML d'ontologies Web*. École Nationale Supérieure des Télécommunications, 19.

Le Boterf G. (1995), *De la compétence, essai sur un attracteur étrange*, Les Ed. d'organisation, Paris.

Müldner T. & Shakshuki E. (2003), *Teaching Students to Implement Algorithms*, Jodrey School of Computer Science, Acadia University.

Muller O. & Haberman B. (2009), *A Course Dedicated to Developing Algorithmic Problem-Solving Skills –Design and Experiment*, URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.6488&rep=rep1&type=pdf (accessed on 6 October 2015).

Noy N. F., & McGuinness D. L. (2001), *Ontology development 101: A guide to creating your first ontology*.

Ouafia G., Abel M. & Moulin C. (2009), *LOMonto: Une ontologie pour l'indexation d'objets pédagogiques*, AFIA platform workshop: Constructions d'ontologies : Vers un guide de bonnes pratiques, 2009, Hammamet, Tunisia.

Psyché V., Mendes O. and Bourdeau J. (2003), *Apport de l'ingénierie ontologique aux environnements de formation à distance*, Revue des Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF), vol. 10, p. 89-126.

Québec Education Program, Secondary Cycle One (2013), URL: http://www1.mels.gouv.qc.ca/sections/programmeFormation/secondaire1/index_en.asp (accessed on 12 September 2015).

Rajagopal H. (2005), *JENA: A Java API for ontology management*. IBM Corporation, Colorado Software Summit.

Uschold M., & Gruninger M. (1996), *Ontologies: Principles, methods and applications*. The knowledge engineering review, 11(02), 93-136.