Towards Ontology Development for Teaching Programming Language

Gopinath Ganapathi, Ravi Lourdusamy, Veeraraghavan Rajaram

Abstract—Several research frameworks are proposed for the development of teaching ontologies. Ontology as a conceptual structure may work as a mind tool for effective teaching and a visual navigation interface to the learning objects. In this paper we have discussed an approach to the practical ontology development and presented the designed ontology for teaching JAVA programming.

Index Terms—Education Ontology, Mind Map, Concept map, Semantic Web Rule Language.

I INTRODUCTION

Knowledge representation and reasoning is an area of artificial intelligence whose fundamental goal is to represent knowledge in a manner that facilitates inferencing (i.e. drawing conclusions) from knowledge. It analyzes how to formally think and how to use a symbol system to represent a domain of discourse (that which can be talked about), along with functions that allow inference (formalized reasoning) about the objects. Generally speaking, some kind of logic is used both to supply formal semantics of how reasoning functions apply to symbols in the domain of discourse, as well as to supply operators such as quantifiers, model operators, etc. that, along with an interpretation theory, give meaning to the sentences in the logic.

Teachers as knowledge engineers are used to work with concept maps, mind maps, brain maps, semantic networks, frames [5],[9],[15] and other conceptual structures. The visual representation of the general domain concepts facilitates and supports students understanding of both semantic and syntactic knowledge. A teacher operates as a knowledge analyst by making the skeleton of the studied discipline visible and showing the domain's conceptual structure. Ontology can be used to represent the domain's conceptual structure. However, Ontology-based approaches to teaching are relatively new fertile research areas. They originated in the area of knowledge engineering [3], [6], [18], which were then transferred to ontology engineering [7], [8], [10].

Date of paper submission is March 6th 2011.

Gopinath Ganapathy is with the Department of Computer Science, Bharathidasan University, Trichy, India. (e-mail: gganapathy@gmail.com).

Ravi Lourdhusamy is with Department of Computer Science, Sacred Heart College, Tirupattur, India. (+91 9443280319; fax: +91 4179 225060; e-mail: raviatshc@yahoo.com).

Veeraragavan Rajaram is with Department of Computer Science, Sacred HeartCollege,Tirupattur,India.(+91 9952598990 +91 04179-244287 e-mail: ragavan.srv@gmail.com)

Knowledge Engineering traditionally emphasized and rapidly developed a range of techniques and tools including knowledge acquisition, conceptual structuring and representation models [1], [14]. Section1 deals with an introduction. In section 2 the related work are discussed to understand the background of the proposed research.

The theoretical issues of ontology engineering are discussed in section 3. An approach for developing teaching ontologies is presented in Section 4. Section 5 proposes the conclusion.

II. REVIEW RELATED WORKS

Several practical approaches for developing teaching ontologies are proposed. The research framework prepared by Tatiana Gavrilova pursues a methodology that will scaffold the process of Knowledge Structure and ontology design is discussed. Moreover, special stress is placed on visual design as a powerful mind tool. The process of developing a practical ontology from the domain of introductory C Programming is described [17]. The automatic mapping of ontology into java proposed by Aditya Kalyanpur creates a set of java classes using OWL ontology. The OWL ontology files created represents an instance of a single ontology class with its properties, class relationships and restriction-definitions maintained [2].

An approach to design and develop teaching ontologies is discussed by Tatiana Gavrilova. The teaching ontologies are used for teaching and learning C programming concepts [13].

JLOO (Java Learning Object Ontology) is a frame work, for organizing learning objects of Java course in an adaptive e-learning environment. The classification in JLOO is based on the Computing Curricula CC2001 of the ACM and IEEE/CS. Using the curriculum as a guideline; the ontology defines the atomic knowledge units (i.e. learning objects) for an introductory course of java programming. The most significant contributions of JLOO are: 1) Defining the atomic knowledge units of introductory courses of Java language, and the relationships among them, 2) Making the knowledge units of JLOO sharable and reusable, 3) Allowing different learning strategies of an e-learning environment to choose dynamically, using JLOO as a guideline, different learning paths, and 4) Making the realization of adaptive learning easy [11].

III. USING ONTOLOGICAL ENGINEERING FOR TEACHING PURPOSES

ISBN: 978-988-19251-5-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The theoretical issues of ontological engineering are discussed in this section by reviewing different definitions of ontology from literature circulated within the field.

A. Ontology Definitions

Ontology is a set of distinctions we make in understanding and viewing the world.

"An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary." [12].

"An ontology is a formal, explicit specification of a shared conceptualization. Conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use, are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group." [16].

Ontology as a useful structuring tool may greatly enrich the teaching process, providing students an organizing axis to help them mentally mark their visions in the information hyper-space of the domain knowledge.

B. Ontology Development

Tatiana Gavrilova, [17] has proposed a 5-steps recipe for developing ontology:

Glossary development: The first step should be devoted to gathering all the information relevant to the described domain. The main goal of this step is selecting and verbalizing all the essential objects and concepts in the domain.

Laddering: Having all the essential objects and concepts of the domain in hand, the next step is to define the main levels of abstraction. It is also important to elucidate the type of ontology classification, such as taxonomy, partonomy, and genealogy. This is being done at this step since it affects the next stages of the design. Consequently, the high level hierarchies among the concepts should be revealed and the hierarchy should be represented visually on the defined levels.

Disintegration: The main goal of this step is to break high level concepts, built in the previous step, into a set of detailed ones where it is needed. This could be done via a top-down strategy trying to break the high level concept from the root of the previously built hierarchy.

Categorization: At this stage, detailed concepts are revealed in a structured hierarchy. A generalization is performed via bottom-up structuring strategy. This could be carried out by associating similar concepts to create meta-concepts from leaves of the aforementioned hierarchy.

Refinement: The final step is devoted to update the visual structure by excluding the excessiveness, synonymy, and contradictions.

IV. DEVELOPING PRACTICAL ONTOLOGY

In this section an attempt to develop ontology for JAVA programming language following the five step algorithm as discussed in Section 3.2 is defended.

A. Glossary Development

The first step in building ontology is collecting information in the domain and building a glossary of the terms of the domain. To build a glossary for teaching introductory JAVA programming course, the terms are generated from two different types of resources: closed-corpus material and open-corpus material.

The closed corpus materials are in the form of lecture notes that are precisely designed for the course. The open corpus materials include several online tutorials in JAVA programming. The terms were extracted from the lecture notes manually by carefully reviewing the lecture handout. The terms from open-corpus material were extracted automatically [4]. Consequently, the automatically extracted terms and manually extracted terms are combined to build a single glossary. The glossary for Java programming language is developed and it consists of 530 words.

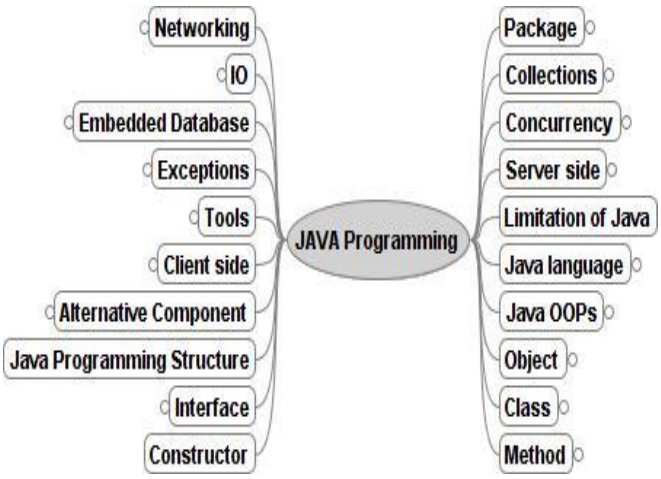

Figure 1 Mind Map

B. Laddering: Building an Initial Mind Map Structure

The second step is to build an initial visual structure of the glossary terms. The main goal of this step is to create a set of preliminary concepts and the categorization of those terms into concepts. A mind map can be a useful visual structure. The mind map developed to design java teaching ontology is presented in **Figure 1**. Since the categorization is preliminary, some of the terms might not fit into any of this initial categorization. The categorization is done manually in this step. However the lecture notes employed were used to build glossary, and to build the initial categorization as well. When designing the ontology the lecture notes are equally compared with experts help.

ISBN: 978-988-19251-5-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

C. Disintegration and Categorization: Building a Concept map with more Precise Hierarchy

The next step is to built a visual structure by analyzing the glossary. First we employed the top-down design strategy to create meta-concepts such as “Class”, “Object”, and “IO”. Then using the bottom-up strategy we tried to fit the terms and concepts into the meta-concept. We have created the relationships between the concepts. (Figure 2)

A concept map is the most useful visual structure for representation of the results of this stage, since it gives the ability of defining the relationship in addition to building the hierarchy. The output of this step is a large and detailed map, which covers the course in a hierarchical way. However, since this ontology is designed for teaching purposes it is important to offer the overall picture and a general hierarchy as well.

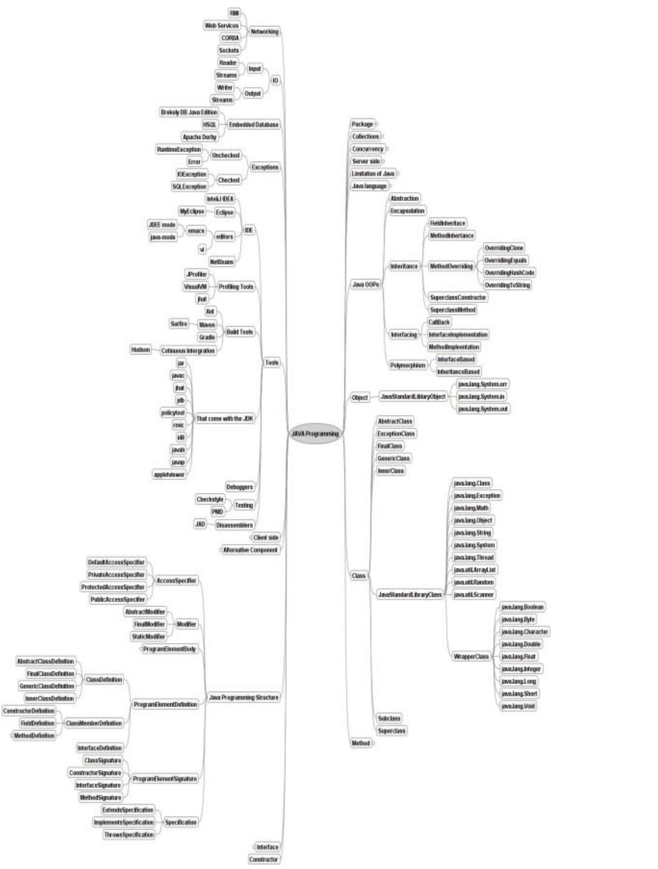

Figure 2. Concept Map

D. Refinement

The general ontology is compared with a refined ontology to update and to get the next version of ontology(Figures 3 and 4). Hence it is an incremental approach. It’s not easy to build all relationship in depth of knowledge. To get the clarity on the ontology developed we have to remove unnecessary node and use the standard relationship that are easy to understand.

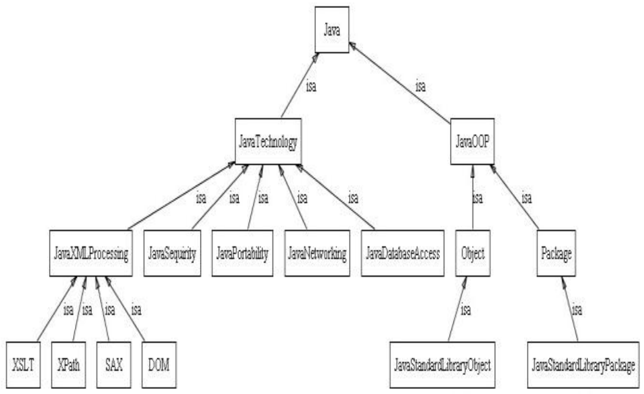

Figure 3. General Ontology

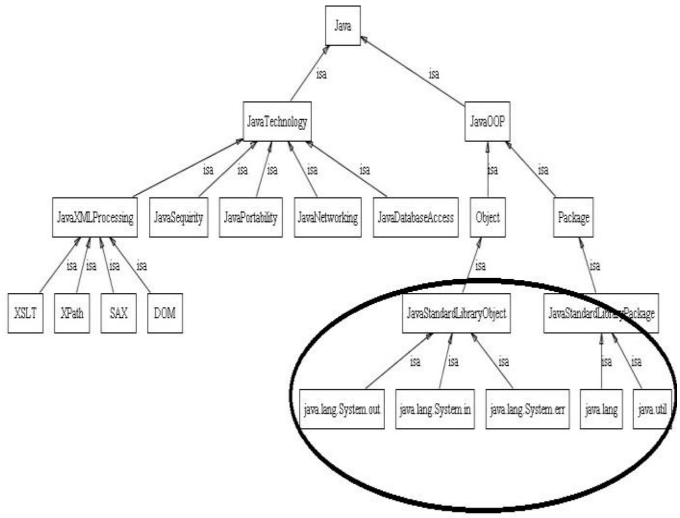

Figure 4. Refined Ontology

V. CONCLUSION

Our java ontology can be integrated with any E-learning platform for class room teaching purposes. The java ontology developed can be further enhanced by adding Semantic Web Rule Language (SWRL), to infer more knowledge

Our research stresses the role of knowledge structuring for developing ontology rapidly, professionally and successfully. The visual paradigm which is used to represent and support the teaching process not only helps a professional trainer to concentrate on problem rather than on details, but also enables a trainee to process and understand great volume of information.

At a basic level of knowledge representation, within the context of everyday heuristics, it is easier for educationalists simply to draw the ontology using conventional “pen and pencil” techniques. However, for more complicated knowledge representations, it is necessary to master appropriate programming and the involved language, or to use well-known ontology editors.

This described approach can be applied to developing those teaching systems where general understanding is more important than factual details. Furthermore, ontology design may be used as an assessment procedure for significant as opposed to exploratory learning. For both formative and summarizing assessment purposes, students can clearly indicate the extent as well as the nature of their knowledge and understanding through creating ontology and explaining the involved processes.

REFERENCES

[1] Adeli, H., "Knowledge Engineering", McGraw-Hill, New-York, 1994.
[2] Aditya Kalyanpur, Daniel Jimenez Pastor, Steve Battle, Juian Padget.,"Automatic Mapping of OWL Ontologies into JAVA",2004.
[3] Boose, J.H., "Knowledge Acquisition Tools, Methods and Mediating Representations. In Knowledge Acquisition for Knowledge-BasedSystems" (Motoda, H. et al., Eds), IOS Press, Ohinsha Ltd., Tokyo,1990, pp.123-168.
[4] Brusilovsky, P. and Rizzo, R., "Map-Based Horizontal Navigation in Educational Hypertext. In Proceedings of Hypertext", University Of Maryland, College Park, USA, 2002
[5] Conlon, T.,"Towards Diversity: Advancing Knowledge-based Modelling with Knowledge Acquisition", In Proceedings of 8th *International PEG Conference*, Sozopol, Bulgaria,1997, pp. 379-386.
[6] Eisenstadt, M., Domingue, J., Rajan, T. & Motta, E.,"Visual Knowledge Engineering". In *IEEE Transactions on Software Engineering*, Vol.16, No.10, ,1990, pp.1164-1177.
[7] Fensel, D., "Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce", Springer, 2001.
[8] Jasper, R. and Uschold, M., "A Framework for Understanding and Classifying Ontology Applications", In *Twelfth Workshop on Knowledge Acquisition Modeling and Management KAW'99*, 1999.
[9] Jonassen, D.H.,"Designing constructivist learning environments", In *Instructional design models and strategies* (Reigeluth, C.M. (Ed)), 2nd ed., Lawrence Eribaum, Mahwah, NJ, 1998.
[10] Mizogushi, R. and Bourdeau J., "Using Ontological Engineering to Overcome Common AI-ED Problems", *International Journal of Artificial Intelligence in Education*, volume 11, 2000, pp.1--12.
[11] Ming-Che Lee,Ding Yen Ye & Tzone I Wang, "Java Learning Object Ontology", Proceeding of the *Fifth IEEE International Conference on Advance Learning Technologies*, 2005.
[12] Neches R, Fikes RE, Finin T, Gruber TR, Senator T, Swartout WR, "Enabling technology for knowledge sharing", *AI Magazine* 12(3),1991, pp.36–56.
[13] Sergey Sosnovsky, Tatiana Gavrilova, "Development of Educational Ontology for C-Programming", *International Journal for Information Theories & Applications*, Vol 13, 2006.
[14] Scott, A., Clayton, J.E. & Gibson E.L.,"A Practical Guide to Knowledge Acquisition", Addison-Wesley, 1994.
[15] Sowa, J. F. ,"Conceptual Structures: Information Processing in Mind and Machine", Addison-Wesley, Reading, Massachusetts, 1984.
[16] Studer R, Benjamins VR, Fensel D, "Knowledge Engineering: Principles and Methods", *IEEE Transactions on Data and Knowledge Engineering* 25(1-2),1998, pp.161–197.
[17] Tatiana Gavrilova, Rosta Farzan, Peter Bursilovsky, "One Practical Algorithm of Creating Teaching Ontologies", Proceeding of the *NBE* 2005.
[18] Wielinga, B., Schreiber, G. & Breuker J., "A Modelling Approach to Knowledge Engineering", *In Knowledge Acquisition*, 4 (1), Special Issue, 1992, pp.23-39.

ISBN: 978-988-19251-5-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)