Development of an Educational Ontology for Java Programming (JLEO) with a Hybrid Methodology Derived from Conventional Software Engineering Process Models

Santhosh John

***Abstract*—“Semantic Web” refers to W3C’s vision of future web. It aims at a web of data, where information is linked up in such a way to be processable by devices and agents. In a wider canvas, semantic web can be seen as a huge engineering solution. Ontologies are playing the vital role in Semantic Web vision for its full-fledged implementation. Though lot of developments happened in this arena of ontology development in line with the implementation of semantic web, the standardization of process models, tools and methodologies are yet to be saturated. Researches in ontology engineering had pointed out that an effective ontology application development methodology with integrated tool support is a mandatory for its success. The researcher in his previous publication, proposed a hybrid methodology for ontology development by leveraging the well proven process models and methods of software engineering. This paper explains the philosophical, engineering aspects of the newly derived methodology and applies the same for the development of a Java Learning Educational Ontology (JLEO). JLEO organizes the learning hierarchy of Java Programming Language suitable to the related modules spread across the curriculum of Middle East College. An appropriate ontology editing tool has been used for the practical development of ontology.**

***Index Terms*—Java programming, methodology, ontology development, semantic web.**

I. Introduction

Formal and sharable Knowledge Based system is the key aspect of Semantic web. Ontologies in particular fulfill the requirements for knowledge representation for the semantic web. The main idea behind semantic web vision is to provide an opportunity to represent information in the web in a way that software agents and systems can understand, manipulate and provide an environments which is more adaptable, personalized and intelligent [1]. Ontology is a broad term including a wide range of activities and complexities playing a pivotal role in information exchange with respect to Semantic Web vision. Ontology is the main building block of Semantic Web vision to provide the information in machine processable semantic models and produce semantically modeled knowledge representation systems. Ontology is the collection of interrelated semantic based concepts based on a finite set of pre-defined terms and concepts of a domain. The visual representation of the generic concepts of a domain best facilitates both syntactic and semantic knowledge [2]. Ontology is the only solution as a common place to interpret the common meanings of the key terms of a domain where conceptual information is spread across two knowledge bases in web. Ontology development is one of the key and significant concerns. There are many reasons behind ontology development to make it a difficult task. One of the major reasons is the non-standardization of methodologies with integrated tool support.

This paper presents an educational ontology for the teaching and learning domain of Java Programming. In Section II, review of related works is furnished to specify the background of work. The philosophical and engineering aspects of the proposed methodology are discussed in Section III. Application of proposed methodology for the development of Java Learning Ontology Development (JLEO) is presented in Section IV. Conclusion and future work are given in Section V.

II. Related Works and Background of Reserach

Until recently, works on accepted practices in Systems and Software Engineering has appeared somewhat disjointed from the area of formal information representation on the World Wide Web [3]. Still obvious overlaps between both fields are apparent and many now acknowledge merit in a hybrid approach to systems development, combining semantic web technologies and techniques with more established development formalisms and languages like the Unified Modeling Language (UML) [4]. Literatures and practices have shown that maturity of software engineering has been established due to the well proven methodologies and their hand in hand support with modeling languages like UML. Though ontology is the back bone of Semantic Web vision, ontology development has got many concerns. Author strongly believes that a standardized methodology with integrated tool support for domain modeling can make a significant difference in bridging the gap between software engineering and ontology engineering. There is no one correct methodology for developing ontology, since there is no one correct way to model a domain [5].

Many approaches have been proposed by researchers for developing ontologies in educational domain. Development of an educational Ontology for C-programming [6] proposed by Tatiana Gavrilova followed a five-step algorithm for visual ontology design. Visual form influences both analyzing and synthesizing procedures in ontology development process. Later an ontology for teaching Java programming was developed based on the five step algorithm proposed earlier [7]. Glossary development, Laddering, Disintegration Categorization and Refinement are the five

Manuscript received September 15, 2013; revised November 25, 2013.

S. John is with the Computing Department of Middle East College, Knowledge Oasis, and Sultanate of Oman (e-mail: santhosh@ mec.edu.om).

DOI: 10.7763/IJIET.2014.V4.419

core phases of the methodology proposed. The published work focused on the knowledge structuring for ontology development which can be applied to teaching systems where general understanding got more weightage than factual details. A framework was presented-Java Learning Object Ontology (JLOO) [8] used as a guideline for the development and organization of learning objects in introductory Java courses in an adaptive learning system. The classification in JLOO was based on the computing curricula CC2001 of the ACM and IEEE/CS. JLOO followed a purpose oriented model for ontology development. The author in his previous publication [9] highlighted the necessity of a standardized methodology for ontology development with integrated tool support. More established design formalisms and languages like UML should leverage the software engineering tools to Ontology engineering. IBM Research group came up with an approach-*Eclipse Modeling Framework-Based Ontology Engineering System* (EODM) by leveraging Model Driven Architecture (MDA) and Ontology Definition Model (ODM), which enable model transformation [10].

III. Proposed Methodology

The primary consideration is given to bridge the gap between software engineering and ontology engineering by leveraging the well proven techniques and process models of software engineering domain to ontology development while deriving a hybrid methodology for ontology development. Both philosophical and engineering aspects of the new methodology have been adopted from standards. The proposed methodology is derived from two well proven software process models, traditional linear waterfall and iterative Rational Unified Process (RUP) [11].

The proposed ontology development methodology's stages are originated from the lifecycle proposed by the Methontology, [12] a methodology proposed for ontology construction by the Foundation of Intelligent Physical Agents (FIPA), which promotes inter-operability across agent-based applications.The engineering behind the Methontology is the intermediate representation in terms of different models such as specification model-semi-formal specification using a set of intermediate representations, conceptual model and a formalized model (e.g., Description Logic Ontology UML Profile) which will be implemented in an ontology implementation language (e.g., Web Ontology Language).

The proposed methodology classifies the phases into three, such as pre-development, development and post development. Pre development stage is concerned with feasibility study of ontology domain that includes assessing the scope of domain, with clear definition of boundaries. Development stage is with the main goal of producing essential model. The last stage is post development where implementation model definition is constructed. Every phase delivers a specific output with the common goal of creating functional component based on ontology that can be used in several systems and scenarios.

The final framework of the proposed methodology fits the various stages of ontology development into the phases of incremental and iterative development methodology, RUP. This provides a disciplined approach to assigning tasks and responsibilities within a development team. The Rational Unified Process captures many of the best practices in modern software development in a form that is suitable for ontology development too. The phases of proposed ontology development methodology along with their stages are fitted into RUP phases, Inception, Elaboration, Construction and Transition. The overriding goal of the inception phase is to achieve concurrence among all stake holders on life-cycle objectives for the project and can map to feasibility study. The purpose of the elaboration phase is to analyze the problem domain, establish a sound architectural foundation, develop the project plan, and eliminate the project's high risk elements. During the construction phase, all components and application features are developed and integrated to the product, and all features are thoroughly tested. These objectives are well mapped to the ontology definition phase. Finally the ontology implementation is mapped towards transition phase. Fig. 1 illustrates the final framework of the proposed ontology development methodology.

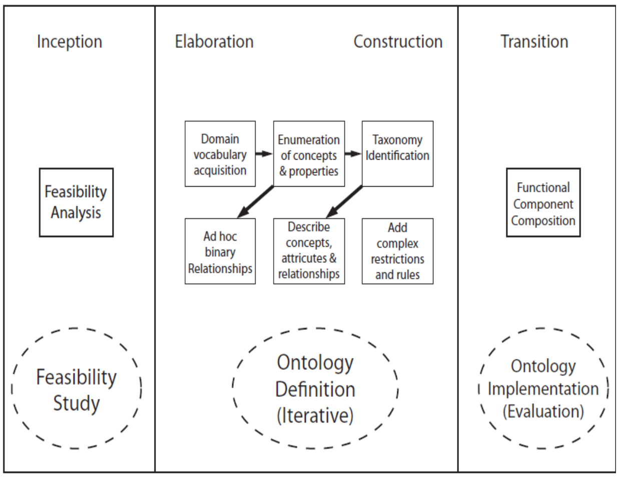
Inception
Elaboration
Construction
Transition
Feasibility Analysis
Domain vocabulary acquisition
Enumeration of concepts & properties
Taxonomy Identification
Ad hoc binary Relationships
Describe concepts, attricutes & relationships
Add complex restrictions and rules
Functional Component Composition
Feasibility Study
Ontology Definition (Iterative)
Ontology Implementation (Evaluation)

Fig. 1. Framework of proposed methodology.

IV. Application of Methodology

The domain chosen for the application of proposed methodology is taken from education field where knowledge sharing and reusability is much required. The growth of e-learning and computer based educational systems should support the import and export of knowledge in a standard format with common semantics. Ontology can be used as a skeletal foundation for a knowledge base that will allow different applications to speak the same language. The proposed methodology is applied for an ontology development for teaching and learning Java programming. One of the motivations to build ontology for Java programming is connected with the attempts to create more effective teaching strategies of an industry language by unifying the different views on the domain. As of now, different teachers introduce Java programming based on many different parameters of their own, such as the order of topics, emphasis on concepts... Introduction of an ontology ensures the uniformity among different views in the domain. Though the order in which the teacher presents the material is upto him/her, the basic hierarchical link structure is not violated. The different stages of the proposed methodology are furnished below in line with the domain chosen.

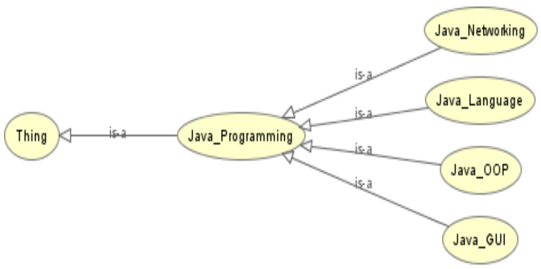

Fig. 2. General structure of ontology.

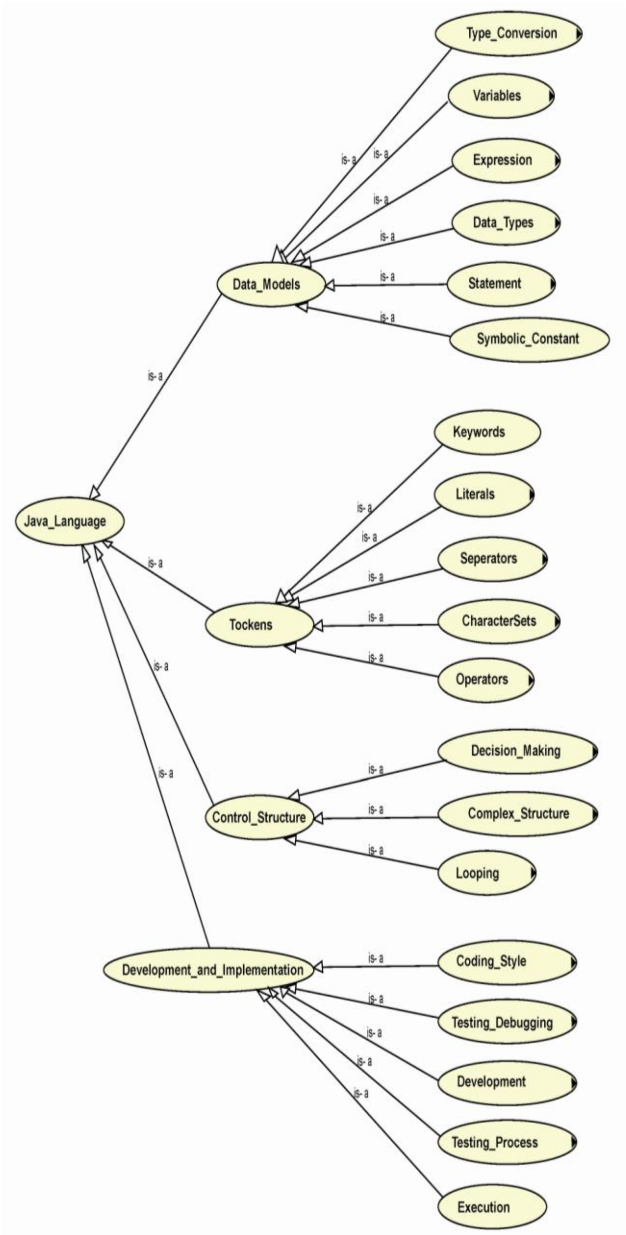

Fig. 3. Taxonomy of Java language concept.

A. Feasibility Analysis

The ontology focuses on the Java 2 Standard Edition (J2SE). The Ontology is derived for the domain of Java learning where Java's approach to variations in Programming is considered. The feasibility of the project is very high where the fully developed ontology after evaluation can integrate with e-learning systems such as Moodle as a guideline, different learning paths and making the realization of adaptive learning easy.

B. Domain Vocabulary Acquisition

For the development of ontology, the concepts/classes are derived from the domain vocabulary acquisition phase. The vocabulary of domain is taken from various resources such as the course materials of the modules - Introduction to Programming, Object Oriented Programming, Distributed programming with Java, Data Structures and Algorithms of Middle East College- Sultanate of Oman, Text books and online resources. Manual and automated mechanisms are applied to extract the terms. The Concepts for the first version of ontology used are the core concepts of Java_ Language, Java _OOP, Java _GUI, and Java _Networking.

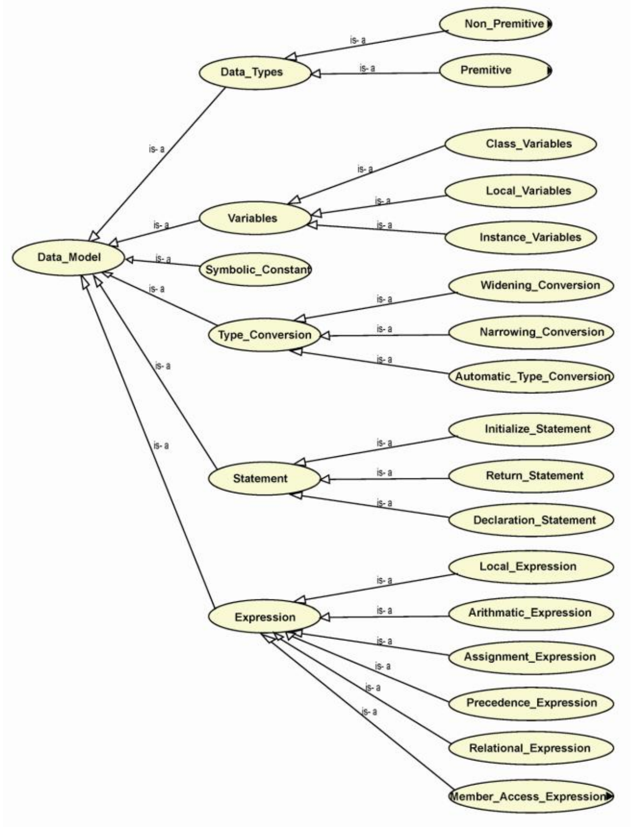

Fig. 4. Binary relationship on data_model concept.

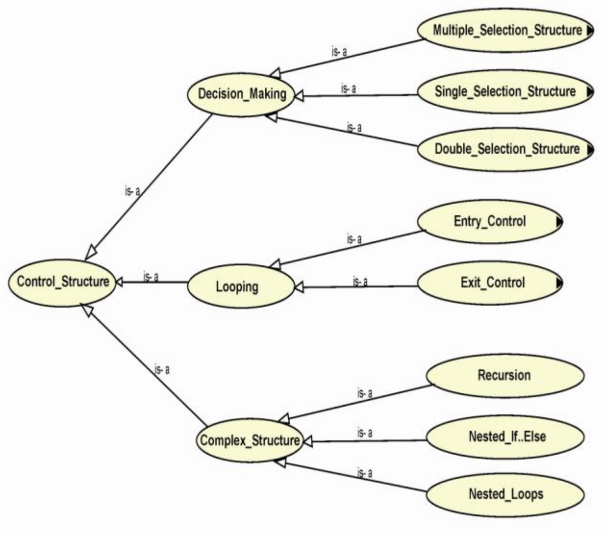

Fig. 5. Binary relationship on control structure concept.

C. Enumeration of Concepts and Properties

This phase produces an enumeration of concepts and properties. A semantic translation among concepts is incorporated here. Properties and property values are assigned at this stage to achieve the shared vocabulary. Prot ég é4.3 has been used to model the phase. Fig. 2 shows the general Ontology structure based on the concepts derived. The subclass hierarchy with appropriate properties is made for the four core concepts/classes done in the phase of Taxonomy Identification.

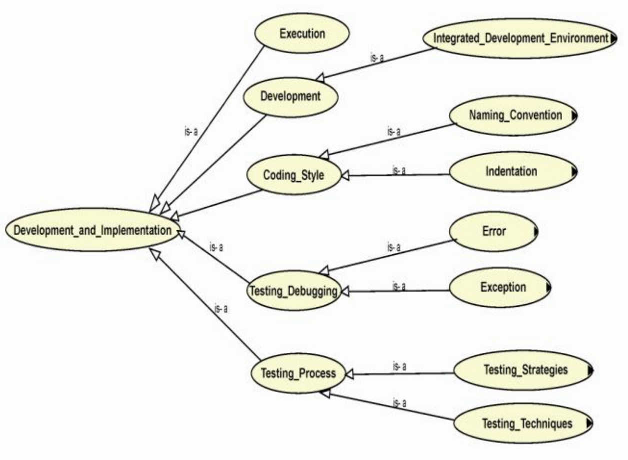

Fig. 6. Binary relationship on Development_and_Implementation concept.

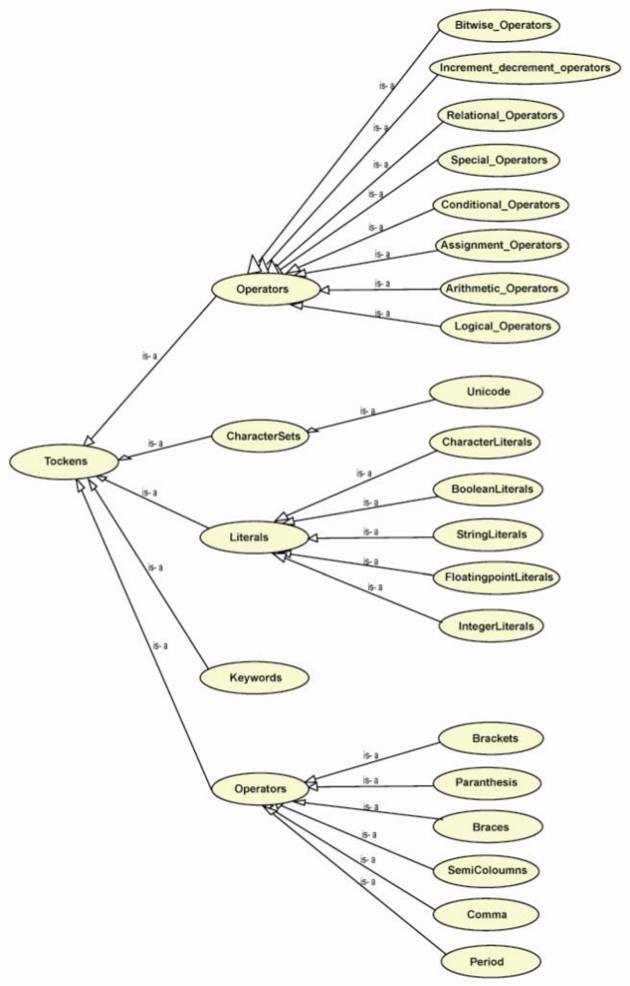

Fig. 7. Binary relationship on Tockens concept.

D. *Taxonomy Identification and Adhoc Binary Relationship*

The core idea behind this phase is the concept hierarchy.

The class hierarchy of the core concepts is the deliverables of the phase. Top-Down approach has been followed where the process starts with the most general concepts of the domain and subsequent specialization of the concepts. Initial version of taxonomy identification of Java Language Concept is shown in Fig. 3. Hierarchies organise the classes into a hierarchical taxonomy by asking, if by being an instance of one class, the object will necessarily be an instance of some other class. If a class A is a superclass of class B, then every instance of B is also an instance of A.

Adhoc binary relationships among the concepts of Data Model, Control Structure, Development_and_Implementation and Tockens belonging to Java_Language are shown in Fig. 4-Fig. 7.

E. *Describe Concepts Attributes and Relationships*

This phase describes the internal structure of the concept. These properties become slots attached to classes. This stage also includes the relationships between individual members of the class and other items. Fig. 8 is a sample screenshot of property values assigned to the concept "**for**" belonging to Entry_Control super class. Both Object property and Data property are assigned.

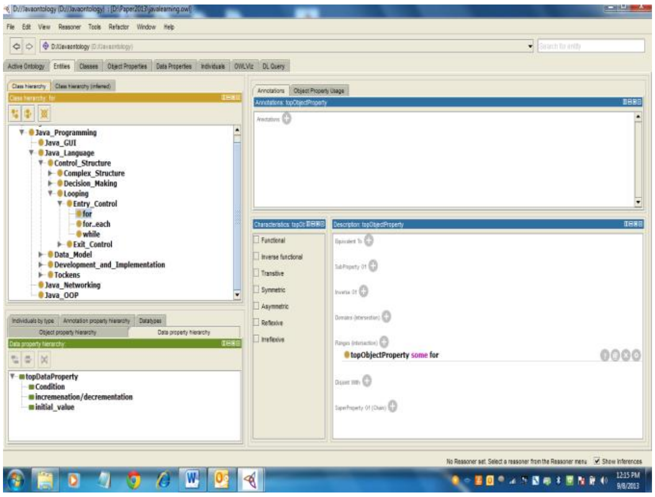
D://Javaontology (D://Javaontology) : [D:\Paper2013\javalearning.owl]
File Edit View Reasoner Tools Refactor Window Help
D:/Javaontology (D:/Javaontology)
Search for entity
Active Ontology Entities Classes Object Properties Data Properties Individuals OWLViz DL Query
Class hierarchy Class hierarchy (inferred)
Class hierarchy: for
Java_Programming
Java_GUI
Java_Language
Control_Structure
Complex_Structure
Decision_Making
Looping
Entry_Control
for
for..each
while
Exit_Control
Data_Model
Development_and_Implementation
Tockens
Java_Networking
Java_OOP
Individuals by type Annotation property hierarchy Datatypes
Object property hierarchy Data property hierarchy
Data property hierarchy:
topDataProperty
Condition
incremenation/decrementation
initial_value
Annotations Object Property Usage
Annotations: topObjectProperty
Annotations
Characteristics: topOb
Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Reflexive
Irreflexive
Description: topObjectProperty
Equivalent To
SubProperty Of
Inverse Of
Domains (intersection)
Ranges (intersection)
topObjectProperty some for
Disjoint With
SuperProperty Of (Chain)
No Reasoner set. Select a reasoner from the Reasoner menu
Show Inferences
12:15 PM
9/8/2013

Fig. 8. Screenshot of property values and complex restrictions and rules.

This stage describes the value type, allowed values, the number of the values (cardinality), and other features of the values the slot can take. For example, the value of a name slot is one string. That is, name is a slot with value type String. A slot produced can have multiple values and the values are instances of the class.

F. *Vocabulary Linking with Data*

The last step is creating individual instances of classes in the hierarchy. Defining an individual instance of a class requires choosing a class, creating an individual instance of that class, and filling in the slot values. At this stage the vocabulary is absolutely linking with the real data.

V. Conclusions and Future Work

This research paper implements a new hybrid methodology for ontology development derived exclusively from well-known software engineering process models. The proposed methodology applied for creating an educational ontology for teaching and learning Java Programming. The developed ontology can be further enhanced by adding Semantic Web Rule Language (SWRL).The derived ontology produces a solid hierarchical structure of Java topics to be incorporated for a standardized ontology for learning Java programming. The derived ontology can be integrated with any E-learning systems for class room learning purposes as domain knowledge representation model. Derived ontology is a sharable and a reusable one for academic institutions and colleges where Java programming is a part of their curriculum and related modules spread across different levels. Visualization of Ontology has been made with the support of an ontology editor. Though the general ontology has a good coverage on Java topics, the current version of ontology is limited to Java_Language concept.

Future work will complete the total ontology development of the domain concerned and will integrate with E-learning system (Moodle). The integration will be experimented in Middle East College for evaluating the viability of developed ontology. In future work, the total evaluation of the methodology will incorporate the possibility of applying the derived ontology for semantic web-the ultimate aim of this research.

ACKNOWLEDGMENT

I would to take this opportunity to express my deep gratitude and gratefulness to Dr. Mary Shanthi Franklin for her continuous guidance, monitoring and constant encouragement to complete this paper through various stages.

I also take this opportunity to express a deep sense of appreciation to Dr. Arun N.S, Head of Computing Department of Middle East College, Sultanate of Oman for his support, and motivation without which this paper would not be possible.

REFERENCES

[1] B. E. B. Costa, M. Silva, and E. Soares, “Acomputational model for developing semantic web-based educational systems,” *Knowledge-Based Systems*, vol. 22, no. 4, pp. 302–315, 2009.

[2] D. Fensel, *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*, 2001, Springer.

[3] Semantic web Activity. [Online]. Available: http://www.w3.org/2001/sw/.

[4] P. Tetlow, J. Z. Pan, D. Oberle, E. Wallace, M. Uschold, and E. Kendall, “Ontology driven architectures and potential uses of the semantic web in systems and software engineering,” *World Wide Web Consortium, Tech. Rep.*, February 2006.

[5] D. Gasevic, D. Djuric, and V. Devedzic, *Model Driven Architecture and Ontology Development*, 1st edition, July 2006, Springer.

[6] S. S. T. Gavrilova, “Development of Educational Ontology for C-Programming,” *International Journal "Information Theories & Applications*, vol. 13. pp 303-308, 2006.

[7] G. Ganapathi, R. Lourdusamy, and V. Rajaram, “Towards Ontology Development for Teaching Programming Language,” presented at the World Congress on Engineering, 2011, London, UK.

[8] C. M. Lee, D. Y. Ye, and I. T. Wang, “Java Learning Object Ontology,” presented at IEEE International Conference on Advanced Learning Technologies, 2005.

[9] S. John, “Leveraging Traditional Software Engineering Tools to Ontology Engineering under a New Methodology,” in *Proc. 2010 5th International Conference on Future Information Technology (FutureTech)*, 2010, pp.1, 5, 21-23.

[10] An MDA-Based Software Development Environment for Ontology Engineering. (2005). [Online]. Available: http://domino.research.ibm.com/library/cyberdig.nsf/papers/0D884EF43842DFD8852570BD0060E974/$File/rc23795.pdf

[11] Bruce MacIsaac. (2003). An Overview of the RUP as a Process Engineering Platform. [Online]. Available: http://www.open.org.au/Conferences/oopsla2003/PE_Papers/MacIsaac.pdf.

[12] O. Corcho *et al*., *Ontological Engineering*, Berlin: Springer, 2004.

Santhosh John is a faculty in computer science since 2000 of premier IT institutions of India, Malaysia and Sultanate of Oman. Completed his masters from University of Madras-India (1998) in Computer Applications and pursuing PhD in Computer science. He published peer reviewed papers in journals and conference proceedings. Grant recipient of Microsoft Knowledge Capital Centre-Malaysia (2003).He teaches software engineering, OOAD, Java programming and Mobile Commerce. His research interests are Ontology development for education, Semantic Web systems for education, Software engineering for education and cross platform mobile application development.