Journal of Computations & Modelling, vol.2, no.4, 2012, 75-91
ISSN: 1792-7625 (print), 1792-8850 (online)
Scienpress Ltd, 2012

Domain Ontology for Programming Languages

Mr. Izzeddin A.O. Abuhassan[1] and Akram M.O. AlMashaykhi[2]

Abstract

Ontology have become a relevant representation formalism and many application domains are considering adopting them. This attention claims for methods for reusing domain knowledge resources in the development of domain ontologies. Accordingly, in this paper we discuss a general methodology to create domain ontology for more than one object oriented language (OOP) like Java, PHP and C++. A lot of software development methods specially Web applications have presented most of these methods that are focusing on the structure of distributed systems and security, in which they are connected through networks and the internet; resulting in more valuable business and critical assets stored, searched and manipulated by World Wide Web.

The aims of this study building domain ontology for OOP language classes for different OOP languages or different versions of the same language is an exciting opportunity for researchers to access the information required under the constant increase in the volume of information disseminated on the Internet.

By creating Ontology domain for OOP, we can Improve methods of viewing and

[1] E-mail: iabuhassan@newsoft.ps
[2] E-mail: akram.othman@gmail.com

Article Info: *Received* : October 14, 2012. *Revised* : November 19, 2012
Published online : December 30, 2012

organizing information, improve the way of processing, in addition to increasing the vocabulary and their relationship to terminology as well as the rules used in natural language with OOP languages. The clear identification of the properties and relations of terms is the starting point to become Ontology domain.
The importance of the domain Ontology among object oriented programming languages is that through the synthesis of these relationships or Ontology an OOP can be achieved through web by any junior programmers.

Keywords: Ontology Domain, Web Ontology Language, Resource Description Framework, OOP

1 Introduction

The Operating system (OS) contains Software (SW) consisting of programs and data, which runs on computers, manages computer hardware resources and provides common services for execution of various application software.

Software programs, computer programs or computer software are programs or routines for a computer system or part of the system itself. Application software programs are often add-ons, specially made for the operating system, and designed to add various types of functionality for the user. Software programs of many different types are made for users with different aims.

A programming language has many types:

[1] Procedural language.
[2] Object oriented language.
[3] Structural Languages.
[4] Logic Languages.
[5] Act.

2 A brief explanation

The domain ontology for OOP language classes for different OOP languages or different versions of the same language is an exciting opportunity for researchers to access the information required under the constant increase in the volume of information disseminated on the Internet.

The brief overview to Object Oriented Programming Domain (OOPL Domain) show as:

- Put the Ontology theory about the OOPL.
- Collect the specification requirements and information about the classes for the Java and PHP languages.[1]
- Build the database of information collected.
- Chose the tools like Portege 4.1 and any other helping tools to build the Domain.
- Building ontology using Ontology Web Language (OWL).And then translate the OWL to high level programming language using Java, PHP or any other programming language.Finally, deploy the domain to the end user.
- The end user can view, update or insert Java or PHP classes through the web domain.[2]

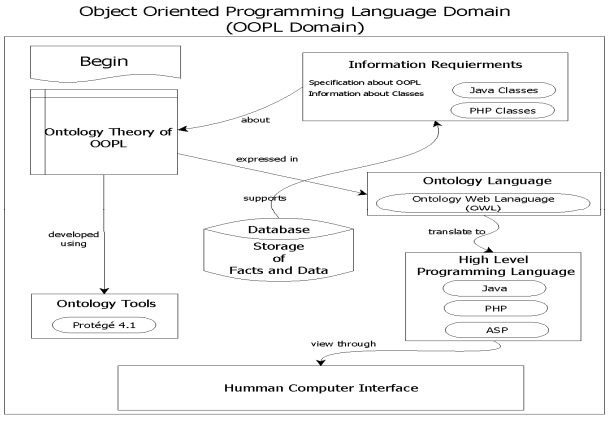

Figure1: Object Oriented Programming Domain (OOPL Domain)

3 Ontology Engineering

In philosophy science, ontology or ontologies in general, is of interest to study what types of things exist, what are the entities that are out there in the universe. Expression derives from the Greek origin meaning "being". A branch of metaphysics is the study of first principles or essence of things.[3]
In the field of information technology, ontology is the business model of the entities and interactions in the field of specific knowledge or practices, such as e-commerce.

In general, it is in the field of computer science and information technology, that we find ontology is a formal representation of a set of concepts within a particular field as well as relations between these concepts.[2][4]

Various Definitions in knowledge engineering, natural languages processing and cooperative information systems, in addition to intelligent information integration, and knowledge management reports about the applications of ontologies in developing and using system in WWW.[8]

In Computer Science Ontology is the formal characterization of a specific domain by defining the concepts of tags and their attributes and relationships between these concepts (relations). Concepts are usually organized according to a hierarchical relationship of globalization and localization between these concepts. In this hierarchical concept there is the offspring of the broader concept of it. For example: "student" is a "human" and "human" is an "Object" (the concept of "student" is the son of the concept "human). All concepts are implicitly members of the concept “Thing”.[10].

4 OOP Domain Methodology

Ontology is used in artificial intelligence, Semantic Web, systems engineering, library science and building information as a form of knowledge

representation of the world or part of it.[4]

The ontology may be as simple as some of the definitions, or may be a division or classification hierarchy, more complex concepts and categories or technology as a solution to the problems based on the significance of information sharing.[8]

In addition to the hierarchical relationship structure of typical taxonomies, ontologies enable cross-node horizontal relationships between entities; thus enabling easy modeling of real-world information requirements. Jasper and Uschold (1999) identify three major uses of ontologies: [12, page 7].

After all of this, we can ask: why are we using Ontology?

The most important reasons are:

- To share common understanding of the structure of information among people and/or software agents.
- To enable reuse of domain knowledge after it exists.

4.1 Ontology and Object Oriented Programming

How can we adapt the concept of OOP to serve of the Ontology?. [5]

Is Object-Oriented Programming is the foundation upon which the concept of Ontologies is built?

This means that we can not get Domain integrated from the beginning, first step building a basic Domain and publishing it in the virtual community, and then the members of the community develop the Domain by increasing Domain information in addition to the amendment to the architecture and its structure.[6]

The main point in this process is that the Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest.[7]

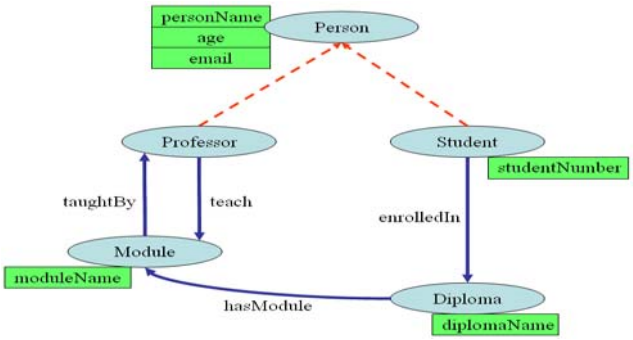

Figure 2: Ontology Relational Concepts

We said previously that the Domain is a terminology and concepts of topics related to or is an expression of Objects in object-oriented programming(OOP) concepts and terminology that can be understood by humans and machines, you can search in your own language the ordinary for any of the subjects related to object-oriented programming (OOP) , so that the agent can understand what you want and return an accurate result, and then any programmer can lead to the development in multi language as Java, PHP, C+++ or any other programming language to become programming in fetcher in natural language.

4.1.1 Ontology Domain Tools

Ontology Web Language (OWL)

There are many languages to describe Ontology. The most recent and best known and most frequently used is OWL.The lifecycle of ontology over the Semantic Web involves different techniques,

Protégé 4.1

Another tool for building Ontology on Web Semantic like using OWL, which is considered one of the most popular open source tools to build Ontology, is Protégé from Stanford University – California.[1]

Protégé 4.1 is chosen to implement this domain ontology because of the fact that it is extensible and : [13] [9]. OWL ontology of gene expression for bioinformatics can be manually built by encoding experts' conceptual patterns, or can be automatically learned e.g. out of a textual corpus by encoding natural language patterns. [12]

The OWL Web Ontology Language is designed to be used by applications that need to process the content of information instead of just presenting information to humans. [6]

RDF and OWL are Semantic Web standards that provide a framework for asset management, enterprise integration and the sharing and reusing of data on the Web. [12]

To Complete Ontology, we need to do several interconnected modules as shown in the figure blow; the core module in this environment is the Ontology services and Ontology Web Language (OWL).The Database Store the data about the domain and the data about ontology.[14]

Ontology services store the likes and html pages about the ontology. Ontology Web Language (OWL) defines the classes and controls the transaction of the domain. Finally, from the web pages, the user can use the ontology as needed. [13]

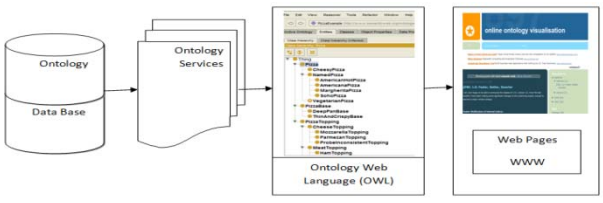

Figure 3: Domain Ontology Output

The UML model is very important to clarify the scope of ontology and to discover levels of the domain as our OOP domain have more than one level as shown in following UML diagram, this diagram covers all basic super classes and sub classes for the programming languages, the first level has super class for

programming languages, the second level has three sub classes from main super class "programming languages", these sub classes are programming languages types. These classes define the types of programming languages, which are the second level in this graph that is the intended programming languages. Languages that are designed, past and present, the other class in this level is "Data types", all types for all programming languages can be under this class and then we can distribute the data types for the languages. The other class in this level is "API's" for all programming languages. We can classify them into groups (super classes and sub classes) [14]. Level three shows the type of programming languages as there are more than one type: structural programming languages, object oriented programming languages, logic programming languages and structural languages.[11] All these types of languages are a notation for writing programs, which are specifications of a computation or algorithm. In addition, the level three has Data types; data types play a central role in all types of programming languages since it enables users to write programs easily and elegantly, different programming languages have been developed, each of which may use different types of data types. Therefore, it becomes very important to organized data types systematically so that we can understand the relationships among one data type and another and investigate future directions which lead us to discover exciting new data types for the same programming language or between different programming languages. [16]

The common data types are Numbers, Characters, Boolean, Set and User defined. Level four of this UML has more details than of the level three; that means Procedural Languages have too many languages some of these programming languages are shown in UML like Perl, Fortran, Visual Basic (VB) and Pascal, other procedural languages are not mentioned in this UML because we are focusing principally on Domain Ontology. In the same level Object Oriented Programming (OOP) has subclasses as Visual Basic.Net (VB.net), C++, Java and PHP. The OOP has specific API's and data types, at the same time it has common

API's and data types. We must demonstrate it clearly in the design of Domain Ontology.

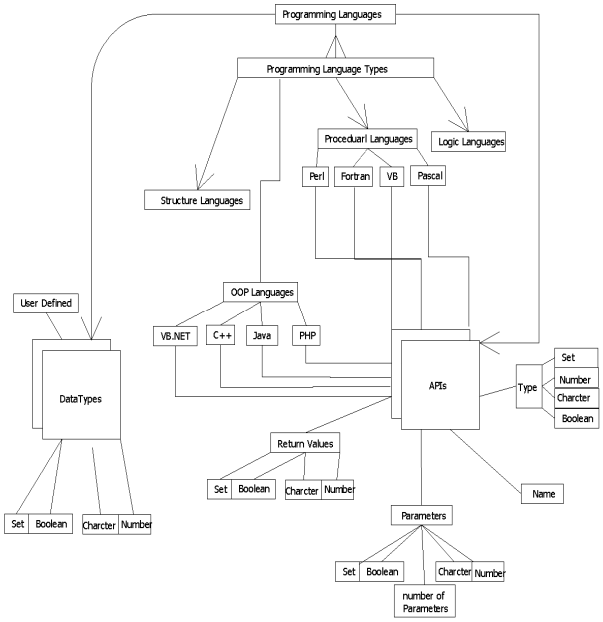

Figure 4: UML for root classes

After all and once the hierarchies and their features have been identified, a table to reflect bidirectional relations may be elaborated by means of assigning names using uniform criteria (or a uniform criterion). Properties may have a domain and a range specified link individuals from the domain to individuals from the range, cardinality and inverse relations. [13][14]

Classes are becoming subclasses-superclass hierarchy, the sup class must be a super class for example OOP is a sub class of Programming Languages at the same time OOP is a super class to Java, PHP, C++, VB.Net; that means that all OOP Languages are sub classes of Programming Languages and Java, PHP, C++ and VB.Net are subclasses of OOP besides all " OOP Languages are Programming Languages and are members of Programming Languages and members of OOP Languages". [15]

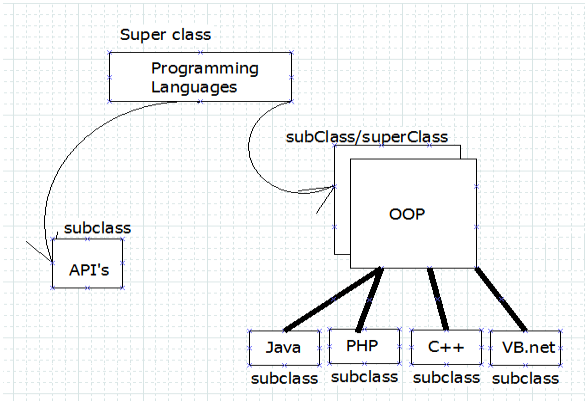

Figure 5: Classes with Taxonomy

The table blow shows the example for class (Programming Languages Classes) and type of relation: one-to-many, one-to-one or many-to-many. Moreover, the relations between classes and types of class:

Table 1: Relations between Programming Languages Classes

Super Class	Relation	Cardinality	Subclass
Programming Languages	Include-PLT	1-n	Programming Language Types
Programming Languages	Include-API's	1-n	API's
Programming Languages	Include-DT	1-n	Data Types
OOP Languages	Include-C++	1-n	C++
Parameters	Include-Number-of-parameter	1-1	Number of Parameter
Java SQL Classes	Include- SQL Statements	1-n	SQL Statements
SQL	Include-	1-1	Statement

Statements	Statement		
SQL Statements	Include-PreparedStatement	1-1	PreparedStatement
SQL Statements	Include-CallableStatement	1-1	CallableStatement
SQL Statements	Include-Connection interface	1-1	Connection interface
SQL Statements	Include-Savepoint	1-1	Savepoint

5 Implementation of Ontology Model/Results

To implement the ontology, we choose the PROTÉGÉ4, because it has tools and it's extensible and provides plug and play environments that can make it more flexible base for rapid prototyping and applications development [13].

Over more to the typical construction of riches ontology. On the other hand the ways in which ontology development environments displays and makes it easy to editing axioms. It is frequently the case that axioms can be rather long. And, justifications can contain long axioms, where only part of the axioms are required for the entailment in question to hold in; in many cases, these parts can obfuscate on the true reasons as to why an entailment holding. Justifications that contain long axioms could also result in information being unnecessarily and poring lost when repairing an ontology, because it isn't clear which parts of the axioms contribute to the entailment explained by the justification. [15]

PROTÉGÉ4 can export to different formats like RDF schema (RDFS) and Otology Web Languages (OWL). We can focus on the problem more easily by using "Reasoner", it's very easy and helpful to locate the error and define it

immediately. Any inconsistence between classes can be defined and corrected [12].

The graph is very good in PROTÉGÉ4 and the view of class and relations in the domain ontology in several ways, graphics using the OWLViz and Ontoviz plug-ins are generated and compared with UML diagrams.

On the one hand, class hierarchies by using OWLViz enable the classes and relations in OWL Ontology to be viewed Graph visualization is a way of representing structural information as diagrams of abstract graphs. The Graphviz layout programs take descriptions of graphs in a simple text language and make diagrams in useful formats such as: images and SVG for web pages, PDF [15]. Graphviz has many useful features for concrete diagrams such as: options for colors, fonts, tabular node layouts, line styles, hyperlinks, rolland custom shapes.

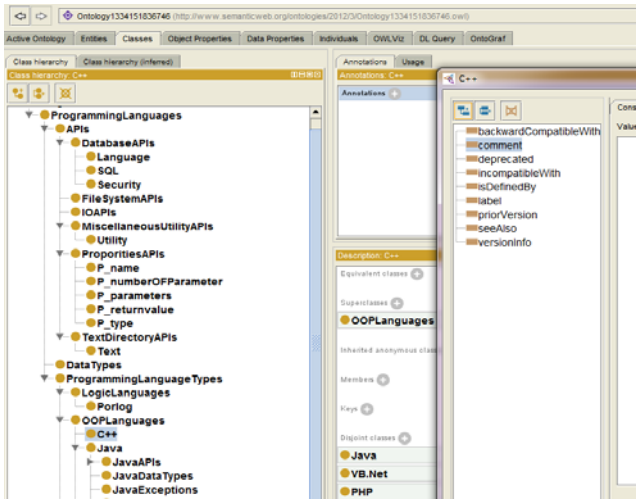

Figure 6: Class Hierarchy

In few words OWLViz computes changes to the class hierarchy may be clearly seen, and inconsistent concepts are highlighted in red. On the other hand, OntoViz generates diverse combinations of graphics with all relations defined in the ontology, instances and attributes. OntoViz allows visualizing several disconnected graphs at once.[8] These graphs are suitable for presentation

purposes, as they tend to be of good clarity with no overlapping nodes, Figure 6 shows the domain inside the PROTÉGÉ4.

To build domain ontology using in our methodology ontology web language with Extensible Markup Language OWL/XML, isthe standard for building Domain Ontology. In addition to these ways, we can implement domain by RDF/XML [9][10]. The below figure shows a side of coding for OOP domain ontology:

Figure 7: Domain Coding

5.1 Main API's Relations

The hierarchy displays all sub classes under the super class Java APIs. In fact the java APIs is more than that viewed in this hierarchy. Every class contains sub classes with more details, the class naming editing to become more closer to the real name classes in java language as java.io its real name in java language that means class in java working with Input/output operations service any transaction input or output from java language.[14] This class can co-operate with other classes and every class has annotation used to add all the information concerning the category. Java language disjoints with other languages (C++, PHP and

VB.Net) so it must be clear in Disjoint classes tab, with OWLViz and OntoGraf the java API's can show as follows:

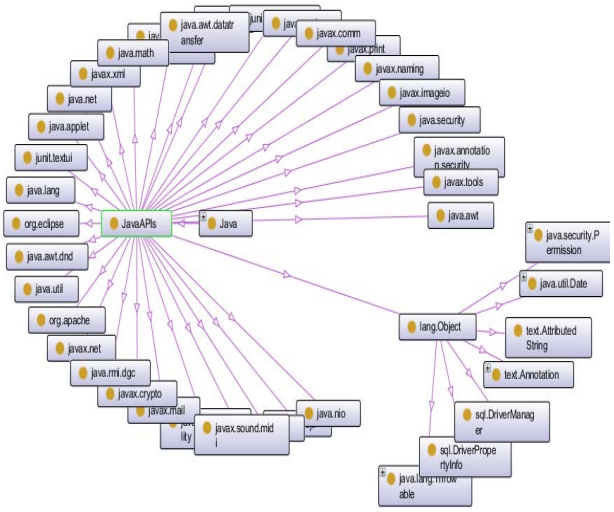

Figure 8: Present API's hierarchy using OWLViz

6 Conclusion

We can Use the architecture and design review to build Domain using all tools of Ontology to enable any person to obtain reach information from Internet. Cannot reach the Web to its full potential if it becomes the source, it can share and process data and information by computers as well as by people.

The computers can be in the process of classification and segmentation and assembling of such large volumes of information or the contents of the Internet to be provided to us as users in the way it is archived and arranged in a way difficult for us to accomplish as human beings and that is exactly what can be done by the Domain Ontology with OWL and other tools.

The OOP Domain Ontology is an exciting opportunity for researchers to access the information required under the constant increase in the volume of information disseminated on the Internet. It can be said that the OOP Domain

Ontology revolutionized the use of browsers in the world of Internet Browsers in the search on the OOP languages.

In the future, the most important part of the Internet will be getting highly reliable and comprehensive information, and is what you will Ontology if they are used and deployed correctly and expanded.

In this paper, we have focused on part of domain Ontology activities that should be an integral part of our Web application life cycle. The future work of this paper will be a research on the translation Services between OOP Languages and ability to use the Ontology Tools like WOL to create a more dynamic domain when retrieves or searches for data and information to increase knowledge field and related experiences.

Finally, there is no single correct ontology-design methodology and we did not attempt to define one. The ideas that we present here are those we found useful in our study.

Acknowledgment: First of all, praise be to Allah, who has helped me complete this thesis, I would like to thank my supervisor Dr. Akram AlMashaiky for his support, patience and good guidance towards a qualitative methodology.
I would like to thank the examining committee, all my colleagues and relatives for supporting me since the beginning. Consequently, many thanks go to all the lecturers and the administrative staff of Amman Arab University.

References

[1] Natalya F. Noy and Deborah L. McGuinness, *Ontology Development 101: A Guide to Creating Your First Ontology*, Stanford Knowledge Systems

Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[2] Dragan Gaˇsevi´c, Dragan Djuri´c and Vladan Devedˇzi, *Model Driven Engineering and Ontology Development, Second Edition*, Springer Dordrecht Heidelberg, London, New York, February 2009.

[3] Loris Bozzato, Mauro Ferrari and Alberto Trombetta, *Building a domain ontology from glossaries: a general methodology*, 2010.

[4] Ontology Modeling and Object Modeling in Software Engineering, *International Journal of Software Engineering and Its Applications*, **3**(1), (January, 2009, 43-60.

[5] Javier Iso, Javier Nogueras Zarazaga Soria and Francisco Javier, *Terminological Ontologies: Design, Management and Practical Applications Lacasta*, Location, Boston, MA, USA, 2010.

[6] Jorge Cardoso, *Semantic Web Services: Theory, Tools and Applications*, The Syntactic and the Semantic Web (SAP Research, Germany), 2007.

[7] Nizar Y. Habash, *Introduction to Arabic Natural Language Processing*, A Publication in the Morgan & Claypool Publishers series SYNTHESIS LECTURES ON HUMAN LANGUAGE TECHNOLOGIES, 2010.

[8] Graciela Brusa, Ma. Laura Caliusco and Omar Chiotti, A Process for Building a Domain Ontology: an Experience in Developing a Government Budgetary Ontology, 2006.

[9] Abid Ali, *Ontology-based Model For The "Ward-round" Process in HealthCare(OMWRP)*, Fareedi, Jönköping University, Master Thesis, 2010.

[10] Grigoris Antoniou1 and Frank van Harmelen, Web Ontology Language: OWL, Department of Computer Science, University of Crete, 2009.

[11] Jorge E. López de Vergara, Víctor A. Villagrá and Julio Berrocal, Applying the Web Ontology Language to management information definitions, *IEEE Communications Society*, (2004).

[12] J. Steven Perry, *Introduction to Java programming*, *Part 1*: *Java language basics*, Object-oriented programming on the Java platform, 2010.

[13] David J. Eck, *Object Oriented Programming using Java*, Object-Oriented Programming School of Computer Science University of KwaZulu-Natal Durban February 5, 2007.

[14] Matthew Horridge, Bijan Parsia and Ulrike Sattler, Explanation of OWL Entailments in Protégé 4, The University of Manchester, Paper, 2007.

[15] Matthew Horridge, Simon Jupp, Georgina Moulton, Alan Rector, Robert Stevens and Chris Wroe, *A Practical Guide To Building OWL Ontologies Using Prot´eg´e 4 and CO-ODE Tools Edition 1.1*, by The University of Manchester, 2007.

[16] Tatsuya Hagino, *A Categorical Programming Language*, Doctor of Philosophy University of Edinburgh, doctorate thesis, 1987.