World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:4, No:4, 2010

Utilizing Ontologies Using Ontology Editor for Creating Initial Unified Modeling Language (UML) Object Model

Waralak Vongdoiwang Siricharoen

Open Science Index, Industrial and Manufacturing Engineering Vol:4, No:4, 2010 publications.waset.org/1529/pdf

***Abstract*—** One of object oriented software developing problem is the difficulty of searching the appropriate and suitable objects for starting the system. In this work, ontologies appear in the part of supporting the object discovering in the initial of object oriented software developing. There are many researches try to demonstrate that there is a great potential between object model and ontologies. Constructing ontology from object model is called ontology engineering can be done; On the other hand, this research is aiming to support the idea of building object model from ontology is also promising and practical. Ontology classes are available online in any specific areas, which can be searched by semantic search engine. There are also many helping tools to do so; one of them which are used in this research is Protégé ontology editor and Visual Paradigm. To put them together give a great outcome. This research will be shown how it works efficiently with the real case study by using ontology classes in travel/tourism domain area. It needs to combine classes, properties, and relationships from more than two ontologies in order to generate the object model. In this paper presents a simple methodology framework which explains the process of discovering objects. The results show that this framework has great value while there is possible for expansion. Reusing of existing ontologies offers a much cheaper alternative than building new ones from scratch. More ontologies are becoming available on the web, and online ontologies libraries for storing and indexing ontologies are increasing in number and demand. Semantic and Ontologies search engines have also started to appear, to facilitate search and retrieval of online ontologies.

***Keywords*—** Software Developing, Ontology, Ontology Library, Artificial Intelligent, Protégé, Object Model.

I. INTRODUCTION

MANY years, ontologies have been implemented in businesses and knowledge base communities as an approach to share, reuse, and develop domain knowledge. Ontologies are now important to many applications such as scientific knowledge portals, information management and integration systems, electronic commerce, and semantic web services. This research introduces the realistic use of the available ontologies which are provided online. The purpose of the research is to transform ontologies to Unified Modeling Language (UML) object diagram using ontology editor "Protégé"[1]. Protégé allows users to build and edit ontologies.

The Author is with the University of the Thai Chamber of Commerce, Bangkok, Thailand. e-mail: waralak_von@utcc.ac.th

Ontologies usually include descriptions of classes, properties (relationship), and their instances. The UML tool such as Visual Paradigm is also used for importing ontology (in XMI format) from Protégé and converts them to UML object model.

"Object diagrams use a subset of the elements of a class diagram in order to emphasize the relationship between instances of classes at some point in time. They are useful in understanding class diagrams" [2]. Object diagram has been widely implemented by the software developing community and its scope is increasing to include more various modeling tasks [3]. This research discusses the practical convergence and expansion of ontologies and UML object diagram and suggests to widely using ontologies in very simple and reasonable way. This research proposes a framework of transforming existing ontologies classes to UML classes and represents it in UML class diagram. The travel/tourism scenario is useful to work out implementation techniques and tool for generating and editing object model as the case study. Ontologies consist of concepts and elements. Developing ontologies represents a great challenge for a number of reasons. It takes a large amount of time and effort to construct ontology, and it necessitates a complicated understanding of the subject domain. Ontologies have been shown to be constructive for representing domain knowledge, and are quickly becoming the backbone of the Semantic Web. Thus it is an even better challenge if the ontology developers are not familiar with the domain. However, one of the major advantages claimed of ontologies is the potential for the "reuse" of knowledge [4]. In theory, one should be able to reuse, modify, expand, and trim other people's ontologies (say "free/available") as required; by this means avoiding the huge effort of starting from scratch. A number of ontology libraries currently exist, hosting various ontology files. Examples of such libraries include Ontolingua, the DAML library, the Protege OWL library, etc. There are an increasing number of online libraries for searching and downloading ontologies.

In order to achieve an effective level of knowledge reuse, we need search engines capable of helping us find the ontologies we are looking for. Some ontology search engines have been developed that can provide lists of ontologies that contain specific search terms, such as Swoogle and OntoSearch [5]. Theses search engines are good step forward, but more is required in terms of ontology search if reuse is to become a reality. Facilitating reuse of other people's

Open Science Index, Industrial and Manufacturing Engineering Vol:4, No:4, 2010 publications.waset.org/1529/pdf

ontologies should encourage more individuals and organizations to participate in the Semantic Web. After all, ontologies are meant to provide an "easy to reuse library of class objects for modeling problems and domains" [6]. Connecting online ontologies might be the first step towards achieving true currently reuse. This paper outlines a useful approach for semi-automatic object model construction. The object model is the beginning phase of object oriented software engineering. The objective of the work is to build a object model from retrieved ontologies according to some criteria. Statement of problem in this research is given in section 2. The connection between ontologies and software developing processes is addressed in section 3. A full description of the architecture and integrating method is given in section 4 and the example of experiments is detailed as well and conclusion will be presented in section 5.

II. STATEMENT OF PROBLEMS

UML was originally designed for building systems in object oriented programming languages [3]. UML is now being used for designing deliverables that are more declarative (similar to ontologies). However, it is time consuming moreover it took a lot of attempt and experience of the software developers to create the UML object model/diagram from scratch. UML practitioners do not usually start a software development by coding. Instead, they have a group discussion, and then they come up with requirement deliverables to carry on for designing phase. "Typical UML include class profiles captured on stacks of physical note cards, diagrams and narratives called use cases that describe how users are expected to interact with the deliverable software product" [7]. UML success is founded on the fact that humans good at communication, express their ideas, bring them to discuss, throw the awful idea and keep the useful one. The production is theses activities is the designing in UML.

There are some similarity between classes in UML and classes in ontologies. This can come with the idea of using them to benefit to each other. In order to creating the object models, we have to find objects which is considered as same as classes provided by online ontologies. Therefore, if there are useful ontologies available, it is superior to use the existing one than building the new. This can support the reusability idea. "The reusability of domain models is often limited because they are domain specific and only takes into consideration abstraction needed to make possible a solution within the confines of their own individual problem space" [8].

How can we transform ontologies to UML object model? We have to find the bridge which can fill the gap between UML and ontology. Ontology usually is represented in Extensible Markup Language (XML), Web Ontology Language (OWL), etc. There are numbers of tools that can be used to derive schemas (XML Metadata Interchange (XMI) to UML) from models automatically such as [11, 12]Visual Paradigm. It is the UML CASE Tool supporting UML 2.1. This research will illuminate all the reasonable steps of transforming ontologies to UML object models in following sections.

III. ONTOLOGY AND SOFTWARE DEVELOPING

Ontologies shows the promising that the Semantic Web can serve as an artifact on which domain models can be produced, shared, and reused. Ontologies are recently initiated tools for structuring knowledge and defined as a collection of concepts (is a class of items) and their interrelationships, which provide an abstract view of an application domain. Simply ontology is an explicit specification of conceptualization [9]. It differs from conceptual models by having declarative data in its model. Ontologies are primarily used in text retrieval. We use ontology to define the semantic content as object ontology, event/concept ontology, and upper level domain ontology [12, 13]. "It is common for software systems to be centered on domain of interesting models. "Domain models can describe the relevant concepts and data structures from an application domain and encode knowledge that is useful to drive an application's behavior. [8]" For example, assume that our task is to develop a system. After some idea, we may come up with an object-oriented design as UML class diagram.

In an ideal world, developers would determine shareable domain models and knowledge parents from a variety of interrelated repositories and then wire them together with the remaining object-oriented components for user interface and control components, a concept becoming recognized as Ontology Driven Architecture [14]. The domain modeling activities (in ontology engineering) in such a development process can be compared to requirements analysis and design steps in traditional software developing [8]. The similarity between ontologies and object models in object oriented software developing is the important issue to address. From researcher point of view, the classes in object oriented are similar to classes in ontologies [15]. Many organizations try to construct ontologies for reusing and integrating. So this benefit comes to the interest of supporting software developing as well. Ontologies are specifications of conceptualizations, used to help programs and humans share knowledge [9]. Ontology provides the fundamental in describing concepts as well as their relationships. It forms the basis in describing how information with a different data structure can be transformed from different application and/or platforms. From a technology point of view, it can be seen as a repository of classes; much like a database represents a repository of data [16].

Ontologies enhance the semantics by providing richer relationships between the terms of concepts/classes [17]. One significant of the major advantages claimed of ontologies is the potential for the "reuse" of knowledge. We should be able to reuse available ontologies, thereby avoiding the huge effort of starting from scratch [18]. There are many online ontologies available for reusing. Some of them are stored in ontology libraries which are storage areas for ontologies. A number of ontologies currently exist, hosting various ontology files. Most of the files are in a machine-readable form [18].

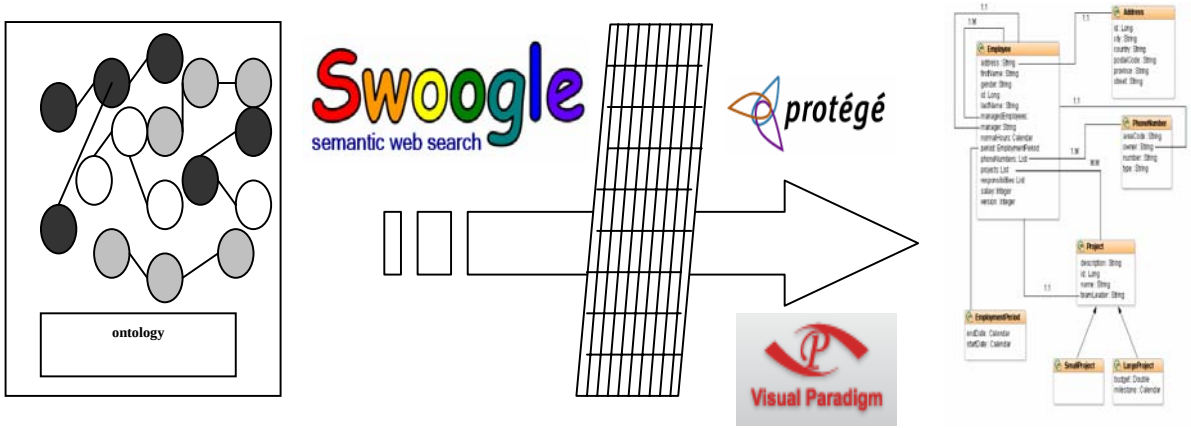


Fig. 1: The Research Framework

There is no one ontology language, therefore ontologies are usually restricted to plain text files or some XML-based storage format [16]. In order to achieve an effective level of ontologies reuse, we need semantic search engines capable of helping us find the ontologies we are looking for. Some ontologies search engines have been developed which can provide lists of ontologies that contain specific search terms or keywords, such as Swoogle (swoogle.umbc.edu). After all, ontologies are intended to provide a simple to reuse library of class objects for modeling problems and domains.

This research indicate the possibilities of using ontologies as the inputs of the semi-automatic object model construction issue which will be advantage for Software Developing. The approach is intended to encourage and support reuse of existing ontologies. The program extracts representations of certain classes of retrieved ontologies, and then allows the developers decide on modification object model components which are attributes, operations and relationships. The result is the visualize UML object model.

IV. RESEARCH METHODOLOGY AND EXPERIMENTAL

A. Methodology

The main research methodology is to use existing ontologies provided in OWL or XML formats. The research will be emphasizing on transforming ontologies to UML object diagram which is the principle of object oriented software developing. A diagram is a partial graphical representation of a system's model.

In Fig. 1 above shows the research framework. The first step starts by searching for available online ontologies using Swoogle. After selected the relevant ontologies, which will usually, come in a different XML-based format language, and they will be chosen in accordance with their relevance of terms and concepts. The different dot-circle box appeared in the left of Fig. 1 represents the different format of ontologies available which are related to each other. Then the program we selected for editing ontologies called Protégé will be utilized to extract the classes/objects and their components and represent classes as the tree hierarchy. Program will have everything we require in order to create object model. However there are many interesting ontologies available, therefore, why don't we use more ontologies. Then we decide to merge two ontologies in order to extract all objects by Protégé and developers will come in the part of editing classes and their components as they decide later.

B. Research Experimental Scenarios

This part is to explain the scenarios of the framework. To help explaining how our proposed system might work, consider the following scenario. Picture this: there is a knowledge developer who is in need of an ontology representing the tourism domain. The ontologies are to be used for creating an object model to hold information on Accommodations, Room, Ticket, Activity etc. There are many ontologies that cover various portions of this domain. It would certainly be beneficial if the developer can quickly and efficiently reuse some of classes in these ontologies, to at least assist the ontology creation. One of the original recommended steps towards building object models is to write down a list of terms to represent as objects in the system. This helps scoping the domain, reaching agreement, and building the class hierarchy. Let's assume that one of the terms that our system wrote was "e-tourism or travel or tourism". There could be many ontologies that covers these concepts to some extent, that our developer is not aware of. It might speed up the task if some existing representations can be easily gathered and represented to accept, modify, or at least learn from [19]. In this case we started with, searching for ontologies on the term "travel" in Swoogle, a list of more than 400 ontologies was returned (see in Fig. 2.).

Open Science Index, Industrial and Manufacturing Engineering Vol:4, No:4, 2010 publications.waset.org/1529/pdf

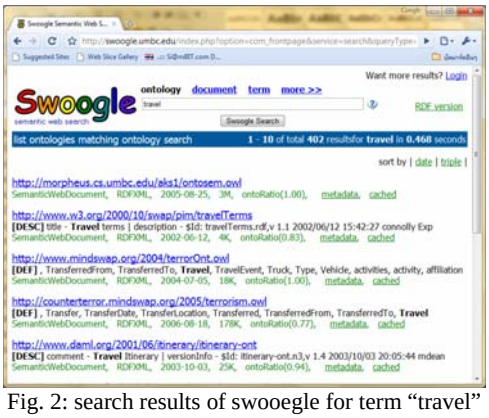


Fig. 2: search results of swooegle for term "travel"

There are too many ontologies available; then we could greatly benefit from an integrating service from the program we chosen ontology editor called Protégé that can extract and edit the classes from ontologies. Then we may decide to take the ontology as a whole, or only take the section. Various organizations rely on UML in the software development process. First step is to find some potentially relevant ontologies to reuse. In our case, the first ontology we selected on Swoogle's results list is "*travel.owl*". This process should produce a list of ontology URIs that is potentially relevant to what ontologies we needs. Let's assume that one of the terms that our system wrote was "*Accommodation*" which is primary interested term in Travel/Tourism domain.

Even though the above *travel.owl* file does represent some concepts, but it might not be sufficient, in terms of coverage, detail, etc. However, many other ontologies were found by the Swoogle for this concept. The second ontology found by Swoogle was "*trip.owl*". then we import slected ontology to editor (in Fig. 3, 4). So we can see our interested in concept/class "*Accommodation*" which is already defined by chosen ontology.

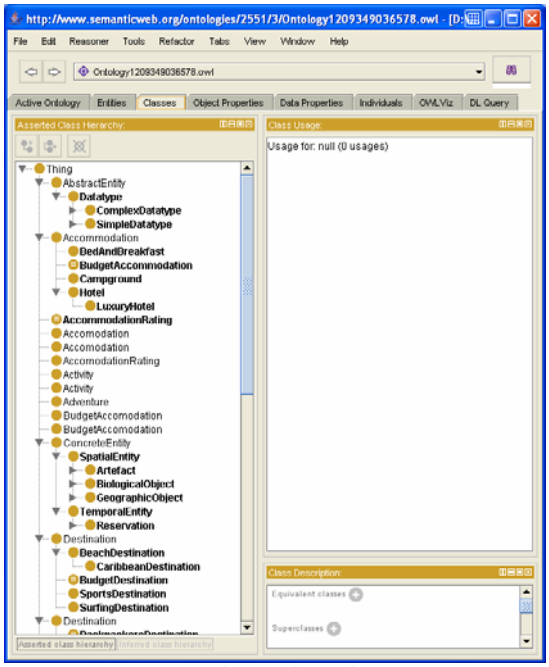


Fig. 3: The ontology classes

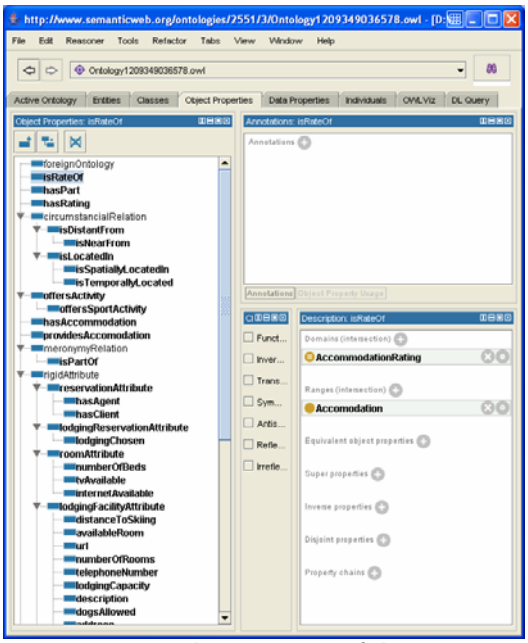


Fig. 4: Object properties of class

In this case we see the classes such as: *Accommodation*, *AccommodationRating*, *Activity*, and etc. The example of Object Properties (Fig. 3) such as " *isRateof*" is object properties that have domain as *class AccommodationRating*, have ranges as class Accommodation. This research sample scenario is attempted to merge two interested ontologies which have the concepts/classes relate with the travel and tourism domain. The merging issue which be applied by using Protégé as well.

After merging both two ontologies into Protégé, the result is to integrate all the components from two ontologies. For example class *Activity and AccommodationRating* come from the *Travel* ontology (http://www.owl-ontologies.com/ travel.owl) and *Airport* from ontology called *Travel Message* ontology (http://www.srdc.metu.edu.tr/webpage/projects /satine/ontologies/TravelMessage Ontology .owl). Here come to editing process of some redundant or unwanted classes or properties or individual in Protégé. Next save file as OWL/XML format. The UML backend plug-in for UML can be downloaded from Protégé official web site. Before opening Protégé, we should installed UML backend plug-in first. Then we import merged ontology in (.OWL) file which we saved from Protégé in XMI format with UML compatible software in this case we use Visual Paradigm. Fig. 6 is showing the results of UML object models in Visual Paradigm.

Open Science Index, Industrial and Manufacturing Engineering Vol:4, No:4, 2010 publications.waset.org/1529/pdf

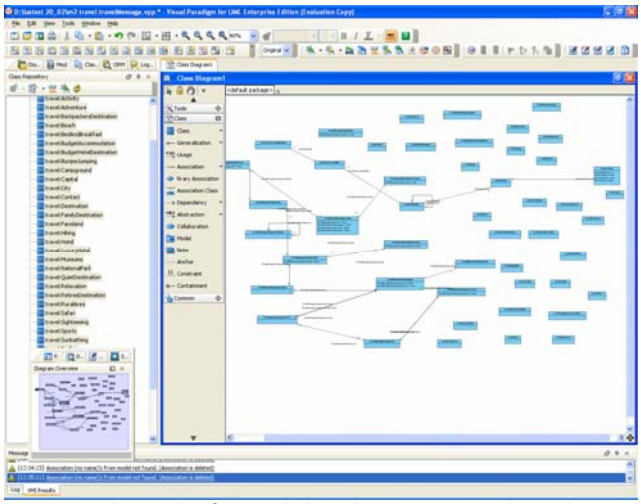


Fig. 6: The result of created class diagram in Visual Paradigm

V. CONCLUSION RECOMMENDATION

This research is reusing existing ontologies; consequently, the quality of those ontologies will certainly shape the quality of the output UML object model as research result. This research presented a new approach for practical semi-automatic construction of UML object model by using the available well known tools. The idea is based on reusing the increasing number of online ontologies to be merged into a new better ontology, rather than the current costly habit of starting from scratch. The proposed system intends to make use of a number of technologies to complete its task, such as ontology searching portal(Swoogle), ontology editor(Protégé), UML tool (Visual Paradigm), these technologies are put together to achieve a common goal.

The research scenario shows how to merge two or more ontologies which is successfully done. Therefore if we need to create more concepts or classes into UML class diagrams, we can also add more ontologies for merging purpose as more as we like which is can be done easily by ontology editor. The experimental result might have some redundancy of classes or properties, which developers or experts can edit/delete/add them before export to UML tools. It will be more functional and comprehensive.

The main successful idea of this research is shown in Fig. 6, it illustrate that we can transform the plain text of ontologies description searched from Internet to UML object diagram as see in Fig. 7. And not only Java class that UML tool such as Visual Paradigm can generate; but also classes are defined in PHP, C#, VB.NET, ODL can be generated as well. For example, we can generate XML schema. On the other hand, UML can be use to create ontologies is possible and promising idea to be recommended to continue in future research.

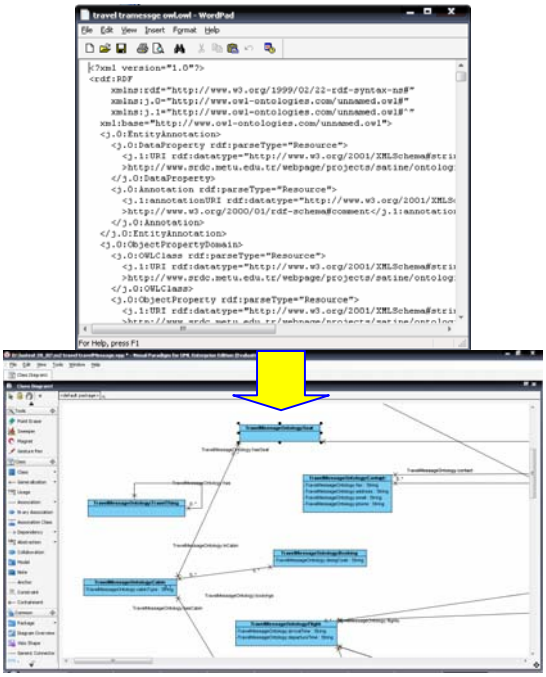


Fig. 7: Research Outcome

Furthermore; Author's recommendation is to create Java classes of all classes extract from ontologies shown in diagram below in Fig. 8 and Fig. 9 by using UML tools. Thus this will be even more convenient to developers to use these Java classes for further software designing process.

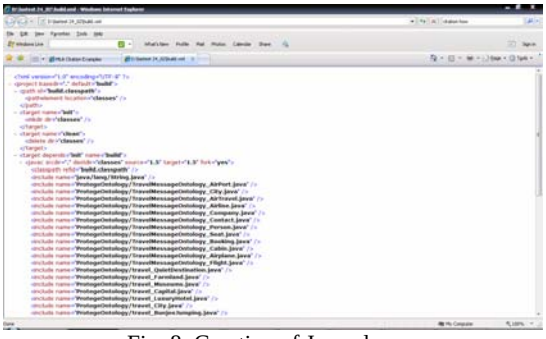


Fig. 8: Creation of Java classes



Fig. 9: All classes in Java files

Open Science Index, Industrial and Manufacturing Engineering Vol:4, No:4, 2010 publications.waset.org/1529/pdf

ACKNOWLEDGEMENT

This research funding is provided by University of the Thai Chamber of Commerce.

REFERENCES

[1] Protégé. (2007) "what is protégé-owl?". Retrieved October 14, 2009. from http://protege.stanford.edu/overview/protege-owl.html

[2] Sparx Systems Pty Ltd., (2009). "UML 2 Object Diagrams", Sparx Systems Pty Ltd., 2009, Retrieved March 25, 2009. from http://www.sparxsystems.com.au/resources/uml2_tutorial/uml2_objectdiagram.html

[3] Kogut, P. and et al., (2001) "UML for Ontology Development". Retrieved March 25, 2008. from www.ccs.neu.edu/home/kenb/pub/2001/05/public.doc

[5] Alani, H., Brewster, C., (2006). Metrics for Ranking Ontologies, EON2006, Evaluation of Ontologies for the Web, 4th International EON Workshop, the 15th International World Wide Web Conference May 22nd, 2006 (Workshop day) Edinburgh International Conference Center, Edinburgh, United Kingdom Retrieved August 20, 2009. http://eprints.aktors.org/554/01/Alani-EON06.pdf

[6] Uschold, M. and Grüninger, M. (2006). Ontologies: Principles, methods and applications. Knowledge Engineering Review, 11(2), 1996.

[7] Laird, C. (2001) "XMI and UML Combine to Drive Product Development. Ideogramic Suite Demonstrates UML-oriented XML Processing". Retrieved March 21, 2008. from http://www.ibm.com/developerworks/xml/library/x-xmi/index.html

[8] Knublauch, H., Oberle, D., Tetlow, P., Wallace, E. (2007). "A Semantic Web Primer for Object-Oriented Software Developers". Retrieved February 20, 2008. from http://www.w3.org/TR/sw-oosd-primer [10] Marchal, E., (2004). Working XML: UML, XMI, and code generation Part 2, The inner workings of UML". Retrived May 20, 2008. from http://www.ibm.com/developerworks/xml/library/x-wxxm24/

[9] Gruber, T. (2007). "What is an Ontology?". Retrieved March 25, 2009. from http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

[11] Marchal, E., (2004). "Working XML: UML, XMI, and code generation, Part 1, Design XML vocabularies with UML tools". Retrived May 20, 2008. from http://www.ibm.com/developerworks/xml/library/x-wxxm23/

[12] Yildirim, Y., Yilmaz, T., and Yazici, A. (2007). "Ontology-supported object and event extraction with a genetic algorithms approach for object classification". Retrieved January 15, 2008. *Proceedings of the 6th ACM international conference on Image and video retrieval CIVR '07*. ACM Press. p. 202-209.

[13] Yildirim, Y., Yilmaz, T., and Yazici, A. (2007) Ontology-supported Object and Event Extraction with a Genetic Algorithms Approach for Object Classification. CIVR'07, July 9–11, 2007, Retrieved August 20, 2009, http://delivery.acm.org/10.1145/1290000/1282314/p202-yildirim.pdf?key1=1282314&key2=0633364521&coll=GUIDE&dl=GUIDE&CFID=55990566&CFTOKEN=43626292

[14] Tetlow, et al. (2005). "Ontology Driven Architectures and Potential Uses of the Semantic Web in Software Developing". W3C Working Draft. Retrieved May 20, 2008. from http://www.w3.org/2001/sw/BestPractices/SE/ODA/

[15] Vongdoiwang, W. and Batanov, D. N. (2005). "Similarities and Differences between Ontologies and Object Model", Proceedings of The 3th International Conference on Computing, Communications and Control Technologies: CCCT 2005, Austin, Texas, USA, 2005.

[16] Siricharoen W. V. (2007). "Discovering Business Object Pattern (BOP) using Integrating Ontologies Library: Tourism and Travel Business Object Pattern Domain". UTCC research paper.

[17] Siricharoen, W. V. (2008) "Merging Ontologies for Object Oriented Software Engineering," ncm, vol. 2, pp.525-530, 2008 Fourth International Conference on Networked Computing and Advanced Information Management.

[18] Ding, Y. and Fensel, D. (2001). "Ontology library systems: The key to successful ontology re-use". In International Semantic Web Working Symposium (SWWS), Stanford, CA, USA. Retrieved November 25, 2008.from http://teacode.com/biblio/er/Ontology%20Library%20Systems.pdf

[19] Alani, H. 2006, Position Paper: Ontology Construction from Online Ontologies,http://eprints.aktors.org/555/01/www06-Alani.pdf

Open Science Index, Industrial and Manufacturing Engineering Vol:4, No:4, 2010 publications.waset.org/1529/pdf