View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by AIS Electronic Library (AISeL)

Association for Information Systems

AIS Electronic Library (AISeL)

ICEB 2009 Proceedings

International Conference on Electronic Business (ICEB)

Winter 12-4-2009

Problem Solving Evolutionary Method for Ontology Knowledge Representation with Protégé-2000

Zhong Yao Dr.

Xiaoxi Chen

Follow this and additional works at: https://aisel.aisnet.org/iceb2009

This material is brought to you by the International Conference on Electronic Business (ICEB) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICEB 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

PROBLEM SOLVING EVOLUTIONARY METHOD FOR ONTOLOGY KNOWLEDGE REPRESENTATION WITH PROTÉGÉ-2000

Zhong Yao[1], Xiaoxi Chen
School of Economics and Management
BeiHang University, Beijing, China
[1] iszhyao@buaa.edu.cn ; [2] Chenxx@buaa.edu.cn

Abstract

This paper studies the knowledge representation with ontology method in the Protégé 2000 system. We first analyzed the various ontological methods for knowledge representation. Then we described the OWL method used in Protégé 2000 for knowledge representation. We proposed the new method named problem-solving evolutionary method (PSEM) for knowledge representation in which it is based the OWL of Protégé 2000. Then we design the interface between the Racer inference engine and the Protégé 2000. Based on the interface built, we can use the Racer inferring engine to reasoning the knowledge. We use the PSEM to experiment the professional domain knowledge of MIS in which it is based undergraduate level. Experiments have shown that PSEM based on the Protégé 2000 is able to represent some domain knowledge well and built knowledge with OWL can be inferred by the Racer.

Keywords: Knowledge Representation, Ontology, Protégé, Problem-Solving Evolutionary Methods, Racer

Introduction

The text as a concept model for knowledge presentation has been an important component in various fields such as knowledge engineering, knowledge management, intelligent system integration, information retrieval, semantic web, etc. In recent years, the ontology has been adopted in many business and scientific communities as a way to share, reuse and process domain knowledge. Ontologies are now central to many applications such as scientific knowledge portals, information management and integration systems, electronic commerce, and semantic web services. According to O'Reilly ＆Associate Inc. statistics in November 6, 2002, there are 52 kinds of the tools for building ontology (also called text editing tools). For example, OntoEdit[5], KAON, WebOnto and Protégé[4], OIL[11], etc. However, these tools are used mainly in manual way to build ontology, in which it leads to time-consuming and needing great effort, easily making preference mistakes and dynamically updating difficulty. Therefore, semi-automatic methods for building ontology had been proposed and developed—ontology learning. The basic concepts of ontology learning are ontology generation, ontology mining and ontology extraction. This kind of work had done in 90s of the last century. There has informed a basic system architecture for building ontology, in which the natural language processing (NLP) and the machine learning are as principle theories[6]. Some systems in corresponding these architectural systems have also been developed such asTextToOntoL3, OntoLeam, the ASIUM system, the Mo'k Workbench, OntoLT, Adaptiva, SOAT and DOGMA etc. The results of ontology learning generally are a draft that requires to confirming by domain knowledge experts before as formal ontology to be used widely. Therefore, the tools of ontology learning often have been integrated into one of typical workable platform for ontology engineering (that is, a system that is a tool for building ontology). Then the system can obtain draft ontology by the ontology learning way to assist the knowledge engineers to build ontology. In this paper, we design a new ontology presentation process based on the ontology building system Protégé. As is well known, the Protégé is a java-based open source ontology editor by Stanford Center for Biomedical Informatics Research. Due to a consistent style with that of MS Windows applications in user interface, it is easy used and learning for common users. The hierarchy of ontology is presented with the tree structure. By clicking items in nodes of tree, users could insert and/or edit class, subclass, properties and instances etc. By that way, the ontology engineers can conceptually design the domain knowledge model without knowing specific ontology presentation languages.

Ontology-based Knowledge Representation Methods

The definition of ontology

Although there are many kinds of descriptions for ontology, it has a basic acknowledge intrinsically that ontology is concerned with static domain knowledge (maybe specific domain, or more widely scope) that as a semantic foundation used for different subjects (people, agent, machine etc.) to communicate (messaging, interoperating, sharing etc.) In other words, ontology will provide clearly defined words lists that can be used to describe the relationship among the concepts so that the users can reach a consistent concept in a specific domain. An ontology describes the concepts and relationships that are important in a particular domain, providing a vocabulary for that domain as well as a computerized specification of the meaning of terms used in the vocabulary. Ontologies range from taxonomies and classifications, database schemas, to fully axiomatized theories [4]. However, because people are not complete understanding the ontology initially, the definition of ontology is developing with the progress of technologies and studies in academia. In this paper, we follow the definition of Studer[9], that is, "an ontology is a formal, explicit specification of a shared conceptualization". This definition is implicitly 4 level meaning[2][3][6]:

- Conceptualization: refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon.
- Explicit: 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, which excludes natural language.
- Share: reflects the notion that an ontology captures consensual knowledge and the concept sets that are accepts as s related domain knowledge, that is, it is not private to some individual, but accepted by a group,

Modeling Primitive of Ontology

López and Pérez organize the ontology with the classification methods and induce 5 basic modeling primitives[7]：

- Classes or Concepts: refers to any events, such as work description, function, behavior, policy and reasoning process etc. From the viewpoint of semantics, it is an objective set. A frame has been used for its definition, including name of concept, relationship set with other concepts, and description of concept with natural language.
- Relations: interaction among concepts in domain, formally, it is an n-dimension subset of Cartesian product: R: $C1 \times C2 \times \cdots \times Cn$. For example, subclass-of. In semantic relationship, it is corresponding to a set-tuple of objective.
- Functions: a class of specific relationship. The nth element can only be decided by the earlier n-1 elements in the relationship. Formally, F: $C1 \times C2 \times \cdots \times Cn\text{-}1 \rightarrow Cn$. For example, Mother-of is a function, and mother-of(x, y) denotes y is the mother of x.
- Axioms: to stand for truth-functionally assertion, e.g. concept B belongs to scope of concept A.
- Instances: stands for element. An instance is an objective in semantics.

Description Language of Ontology

OWL（Web Ontology Language）has been appeared in 2001[2]. The aims of OWL developed by W3C Web-ontology woking group is to provide a kind of language that can be used in various application languages. OWL is W3C organization recommended ontology description language in 2004. It is a semantic markup language that used to distribute and share ontology in WWW. OWL is developed on the basis of DAML+OIL (DARPA Agent Markup Language + Ontology Inference Layer) as an extension of RDF (Resource Description Framework). Its aim is to provide more primitives so that it can support more rich language representation and reasoning. W3C also published a draft framework for web service executed on OWL in order to provide cases and solutions for next generation Web service. According to different demands, OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL (Description Logics), and OWL Full.

Conclusions Problem Solving Evolutionary Method for Ontology Presentation

Problem-Solving Evolutionary Method is originally coming from the practical application of information technology. The aims of introducing ontology to information technologies are for the computers to understand the semantic information in knowledge domain and provide more intelligent service for humans. This process, in fact, is also a software producing procedure. Therefore, software engineering method can be used as a reference for building ontology in information technologies.

In process of software development, four kinds of activities are often conducted: making specification of software, software coding, software confirmation, and software evolution. In corresponding for this process, the basic activities in the procedure of building ontology including: 1）Planning, this activity includes requirement analysis, domain determination, ontology specification making (aim, scope, method and use of ontology); 2）building ontology: knowledge acquisition, analysis, conceptualization (concept abstract, vocabulary determination), formalization (coding), integration, documenting; 3）Confirmation and evaluation: refining and confirmation, assessing correctness and validity; 4）maintaining and evolution. We find that this process is consistent with that of problem-solving method used in software engineering. This process has been noted by Fernández-López et al [2]. In the ontology research, Eriksson et al. [10] proposed a framework for tasking modeling with problem solving method. However, they did not consider the evolutionary characteristic in problem-solving method. In the next sub-section, we introduce the problem solving methods with integrated evolutionary process.

PSEM

By our aforementioned analysis, we propose a new building ontology method, called problem solving evolutionary methods (PSEM). The framework is shown in Figure 1. Note that this figure is different with that of Yu et al. [1] presented in that we have integrated the evolutionary process in ontology knowledge refining process. The PSEM has the following steps to build ontology:

1) Requirement analysis. In this step, we need to clarify the objectives, scope, application and users for ontology to be built. On the face of it, building domain ontology is served for a machine (computer), and also it should be understandable for the machine. However, on the final aim, the ontology built is serving better information for human being. Therefore, similar to the process of software development, in the initial process of building ontology, we should understand and determine the application background and specific requirements. Generally, we can answer the following questions to clarify the demand: which domain knowledge will we build ontology? What kind of application does it for? Who are the target users? How long time does it need to build ontology? Which kind of describing language should it be choose?

2) Planning. In the second step, based on sufficiently understanding available resources and requirement specification, the ontology builder will work out the project specification (proposal). The proposal will include objectives, methods, task allocation, and time demanded of project development. This proposal is quite necessary in practical applications. However, in some project it is often omitted by developing team. Failure implementation in many projects is because they lack a project guide or proposal.

3) Knowledge collection and acquisition. First, developer should understand domain knowledge via various resources collecting knowledge, and the resources include experts, books, internet, journals, magazine and some others. To obtain information or knowledge, the methods may be brainstorming, interviewing, questionnaire, and internet survey such as knowledge automatic acquisition tools.

4) Key concepts and relations determination. After understanding domain knowledge sufficiently, knowledge workers will abstract key concepts and relation between these concepts in some domains and represent it with natural language accurately. Then these concepts and relations will be confirmed by the domain experts. Confirmed knowledge will be as core concept set in domain ontology. Because the experts has different concept and understanding for some concepts and knowledge workers also have different levels, this process can not be sure that all core concept and relation are key concepts and covers all domain knowledge. However, they should satisfy at least:

(1)Determined key concepts and its relations: the relationship among the concepts must be boundary of specific domain and often existing fuzzy relations, the relationship degree are not easy to determine. However, the relation degree or level should be explainable at least.

(2)Be accurately expressed with terminologies: The aim of building ontology is to provide a semantic standard for domain knowledge to be exchanged. Therefore, terminologies should be elaborately selected. It can not be too colloquial and also not too obscure (hard to understand). The basic requirement is that the terminology is unambiguous.

We refer to Wikipedia with bottom-up methods to build the key concepts and its relationships about the knowledge structure in major of information system. In this step, the developer should design a set of terminology, in which it should explain the procedure of selecting terminology and to describe each terms with natural language.

(3)Define the properties of concepts and facets of concepts: if an ontology system only defines a concept system, it can not provide solutions for dynamic requirement for knowledge understanding. Therefore, it is necessary that we need to design internal structure or hierarchy among the concepts once we have set up concept systems. One property is consisted of many facets, which includes value type, allowed values, cardinality and other attributes of related property.

5)Coding

The final objective of building ontology is to understand real life of human by a machine. Therefore, the terminology used in ontology must be coded with formal language. Logic presentation is used more often in research fields. Description logic is a formal representation for ontology knowledge, and it draws the idea from KL-ONE system and a judging subset of first order predicate logic. However, it also have some difference with that of first order predicate logic in that the describing logic system can provide judging reasoning service. Among many knowledge presentation ways, formal ways have been interested in developers and researchers is because they have distinct theory foundation. It is quite suitable for representing applications by conceptual taxonomy and provides a lot of reasoning services. Currently, popular ontology describing language often uses those of logic description languages. In this paper, we use OWL OWL (Ontology Web Language) released by W3C in Feb, 2004 as describing language to representing the MIS filed domain knowledge. After ending the coding process, the codes and documents has to store in file forms so that it can be as standard documents for sharing.

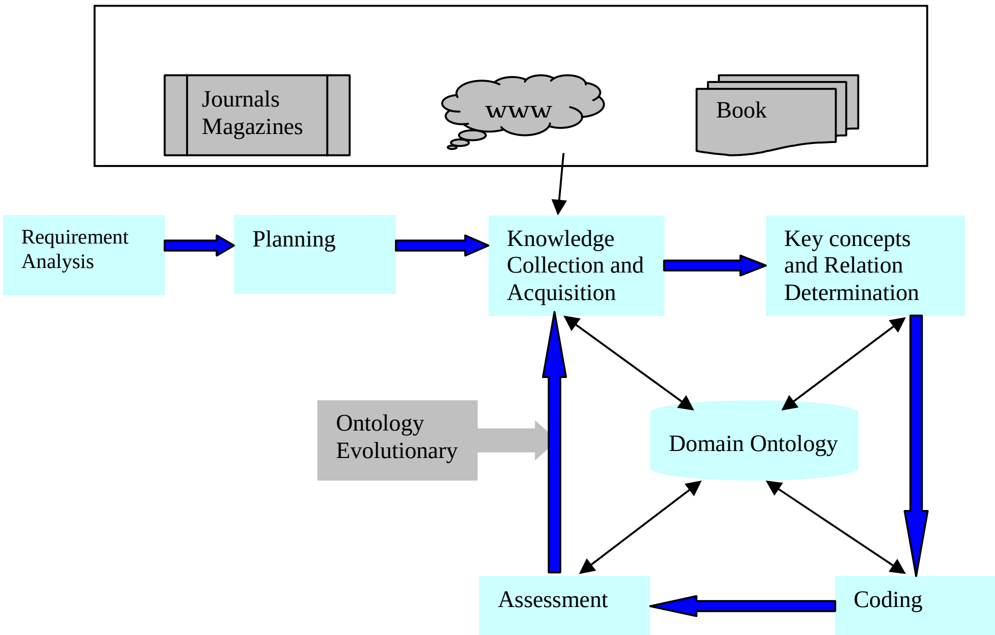

Figure 1. A Diagram of PSEM

6) Evaluation

After above five steps, we have built a basic ontology core. As done in software testing, the built ontology also needs to be tested and assessed. Currently, there is no standard commenting method for ontology, so there is no standard testing set. Comprehensive reviewing evaluation method for ontology, we have following indexes for assessing the ontology: correctness, consistency, expandability, validity, scope of ontology and ability of ontology description.

7) Evolutionary

As is well known, one specific domain knowledge is very complex and its boundary is fuzzy. Domain knowledge is always cross among different domain. Therefore, it can not be built a complete ontology for domain knowledge in one-time, even it is a huge organization and have enough capabilities. In particular, for research organizations it is unreality to build a complete ontology. However, if one organization do one domain ontology development work with iterative process, or by evolutionary process, it can hope get a good ontology system. In each iterative process, the system will be recognized and evaluated by experts and users. If necessary, one can repeat the last time process. The ontology system will gradually expand and evolve into a relatively complete system.

The patterns of ontology evolution can be integrated into new ontology, complementary concepts and relations by experts and discovery new knowledge by machine learning. In addition, in semantic web building process, editing and distributing information requires the workers who know and understand professional knowledge. These workers know domain knowledge ontology building status and manually or semi-automatically use the ontology to label common page semantics. In label process, they find new concepts and relations. If we integrate the label process into ontology building process, we can easy realize the ontology evolution. This evolutionary process is more easily operate than in case of expert definition, and more accurate than the machine learning. However, this process requires an adaptable tool to support.

Reasoning with Racer Inference Engine

After the ontology knowledge has been constructed, the next development is to reason the knowledge and recommend more personalized knowledge to users. We integrate the Racer inference engine into Protégé system to make the knowledge reasoning. The basic framework is as shown in Figure 2.

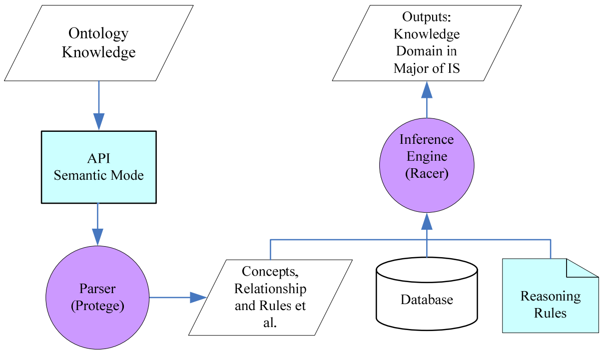

Figure 2. Logic Interface between the Protégé and Racer

1) Analyzing Semantics Model

Here the semantic model analysis means that we first get the factors, such as concepts, rules and relationships, those used to describe semantic models from the invoking API of Protégé. Note that in programming development it needs importing the developing package of Protégé, "import edu.stanford.smi.protege.model.*;"

(1) Accessing Protégé project

Two kinds of ways can be used for this aim: one is directly integrating the Protégé into current project. For example,

```
Project project = ProjectManager.getProjectManager();
KnowledgeBase kb = project.getKnowledgeBase();
```

Where, a project defines a Protégé project that is being used. Kb defines the knowledge base of current project. Based on this knowledge base, other elements can be obtained by invoking Protégé developing package interface. Another way to access the Protégé project is by designated saving path of Protégé building a project to analyze the model. The coding segment is as follows

```
Private static final String PROJECT_FILE_NAME=
        "e:\\dxd\\pprj\\luoding.pprj";
        Collection errors = new ArrayList();
        Project                    project                    =new
Project(PROJECT_FILE_NAME, errors);
KnowledgeBase kb = project.getKnowledgeBase();
```

Where, the string used to input the project directory path, and errors are the message string that error path met. After the project has been obtained, the corresponding knowledge base also has been built.

(2)Getting the elements of knowledge base

Next step is to obtain the elements of the knowledge base through the API provided by the Protégé. These elements include Class, Slot, and the relationship between the Class and the Slot. In Protégé semantic model, rules and relationship (except the farther-child node) all are described with Slot. Then we use the recursion process to analyze whole knowledge base to obtain the tree structure and its root nodes of all class. At the same time, we also get the slots in the knowledge model. The attributes of various concepts are existed in slot form of the protégé. By the way, the slot has a mapping with the class. However, the analyzed slots are attached in related class.
Invoking method such as,

```
getDirectSubclasses(), cls.getDirectSubclasses()
```

(3) Connecting to the database

The information of database are related with the semantic model and mapping. We can get the some concepts database model by analyzing the knowledge model. The database information includes database IP address, type, database name and table name. All these information are attached on the Slot as the OwnSlot.

2) Using Racer to reasoning

Racer system proves a interface with the protegesystem. By sending a message to Racer system, and then receive the results by the Racer reasoning. First it needs to monitor the port of Racer (default 8088), that is to check whether the Racer has started or not. If started, invoking interface provided by the Racer system to build the knowledge base. Then with RQL to search the knowledge base. The connecting operating is as follows:

```
RacerClient                client                =new
RacerClient("192.168.0.88","8088");
Client.openConnection() ;//connectted to Racer
Client.closeConnection() ;//interrupt to Racer
```

Next, input the user's index words into Racer to reasoning. Receive the results of Racer output. Coding segment is as follows.

```
StringBuffer buf = new StringBuffer():
Buf.append(connectClient().synchronoousSend(queryTe
xt));
Buf.toString();
```

3)Analyzed model input Racer.

In our example, reasoning rules are as follows.
(in-knowledge-base MIS)

```
(signature: atomic concepts (business information
management, supply chain management, project
management, organizational culture, database,
management information systems, industrial
engineering, …)
            :roles (( know: person knows person))
                 ((interest: person interests in knowledge
                         : domain person
                         : range professional knowledge
in business information management
```

: roles ((need : person needs knowledge
: domain person
: range professional knowledge in business information management
....
: attribute((knowledge scope: String))
((knowledge domain: String))
((knowledge use: Sting))
:individuals((Zhang Ming，Wu Kaixuan，Liu Simeng ……))
:rules ((Tableaux Algorithm: transitive t))
((axiom: transitive t))
((model analyzing way：transitive t))

This rule defines the concepts, rules and properties of ontology, and builds the algorithm and users personalization. Then, the Racer conducts the reasoning and gets a tree structure in which it is of the hierarchy of structure. In the process of reasoning, Racer also stores the properties of various classes, including the range and domain. When users ask the question to Racer, the Racer accepts this question into Abox, and the matching these concepts with the those stored in Abox. After the end of this matching process, matched data will be transferred into the Tbox. Tbox represents the inclusion relationships. By the data process, fitted results will be output. This completes the one cycle of reasoning.

An Experimental Example

We use the knowledge in management information systems as our domain knowledge to build an ontology. Following above steps to build MIS ontology and then present it in Protégé 2000. First we analyze the preliminary knowledge and professional foundation courses in MIS, then comprehensive main professional courses. On the other hand, we investigate the requirement knowledge in practice, particularly in IT enterprises, we build the knowledge ontology with OWL and then exhibit it on protégé 2000. See Figure 3. (due to room of this paper, we don't introduce this example in detail). Then we use the Racer to reasoning the knowledge built with the protégé.

Conclusions

In this paper, we propose a problem solving evolutionary method to build domain knowledge ontology. We use protégé 2000 as tools to build ontology of domain knowledge in management information systems. Experiments have shown that our new method can effectively improve the ontology building process and provide a convenient process for sharing knowledge. Built knowledge ontology also provides a foundation for the knowledge reference. Therefore, the next step of our work will be design a knowledge reasoning mechanism to refer in above knowledge ontology.

Acknowledgement. This research is partially supported by Natural Science Foundation of China with Project No. 70672020, and 70521001.

References

[1] Yu Shu-Hao, Su Shou-Bao, Liu Ren-Jin. (2007), "A Comparative Studies in Ontology Modeling with UML and OWL," *Journal of Computer Technology and Development* (In Chinese). Vol. 17, Issue 1, pp. 155-157.

[2] Fernandez, Mariano. (1999), "Overview of Methodologies for Building Ontologies," *Proceedings of IJCA1999's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends*, pp. 4.1-4.13.

[3] ISI Natural Language Processing Research Group, (2007), "Ontology Creation and Use: SENSUS," Hhttp://www.isi.edu/natual-language/resources/sensus.htmlH, accessed in 2007-06-01.

[4] The Protégéproject. Hhttp://protege.stanford.eduH, 2002.

[5] Sure Y., Erdmann M., Angele J., Staab S., Studer R., and Wenke D. (2002), "OntoEdit: Collaborative Ontology Engineering for the Semantic Web," In *International Semantic Web Conference 2002* (ISWC 2002), Sardinia, Italia, 2002.

[6] Chen Jie, Yang Dong-Qing, Zhang Ming, Tang Shi-Wei, Deng Zhi-Hong (2002), "Overview ontology," *Journal of Peking University* (In Chinese)，Vol. 38, Issue 5, pp. 728- 730.

[7] Fernández-López, M.; Gómez-Pérez, A. (2002), "Overview and Analysis of Methodologies for Building Ontologies ," *Knowledge Engineering Review*, Vol. 17, Issue 2, pp. 129-156.

[8] Gomez-Perez A. and Benjamins, V. Richard, (1999), "Overview of Knowledge Sharing and Reuse Components," In *Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods* (KRR5), Stockholm, Sweden, August 2, 1999, pp. 1.1-1.15.

[9] Studer R, Benjamins V R, Fensel D. (1998), "Knowledge Engineering, Principles and Methods," *Data and Knowledge Engineering*, Vol. 25, Issue 1, pp.161-197.

[10] Eriksson H, Shahar Y., Tu S.W., Puerta A.R., and Musen M.A. (1995), "Task Modeling with Reusable Problem-Solving Methods," *Artificial Intelligence*, Vol. 79, pp. 293-326.

[11] Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D. L. and Patel-Schneider, P. F. (2001), "OIL: An Ontology Infrastructure for the Semantic Web," *IEEE Intelligent Systems*, Vol. 16 Issue 2, pp.38–45.

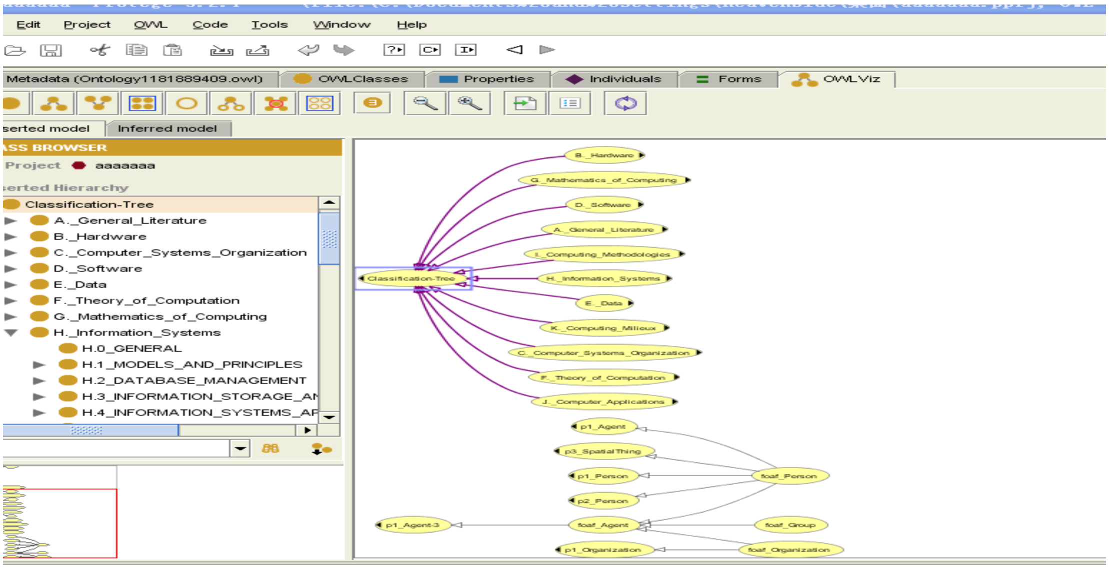

Figure 3. MIS Ontology based on Protégé 2000