Towards Domain-Centric Ontology Development and Maintenance Frameworks

Faezeh Ensan and Weichang Du
Faculty of Computer Science, University Of New Brunswick, Fredericton, Canada
{*faezeh.ensan, wdu*}*@unb.ca*

Abstract

In this paper, we attempt to study and investigate ontology development and maintenance frameworks from a domain-centric point of view. By frameworks we mean the structures which have been designed to allow ontology engineers and domain experts to develop and maintain domain ontologies. Such frameworks usually specify particular phases for developing ontologies and provide implemented components for each phase. Our purpose is to analyze the suitability of a framework for developing ontologies which can fulfill the necessities of a specific domain. We have designed a comparison model for analyzing ontological frameworks. Using the model, we inspect how an ontological framework utilizes domain information resources for creating and maintaining ontologies, how much fineness and granularity the designed ontology can reach, and with how much maturity it supports the maintenance and integration capabilities in the development process.

1. Introduction

In his seminal paper, Gruber [6] defines an ontology as 'an explicit specification of a conceptualization'. Ontologies are used by applications, domain experts and users in order to reach consensus on various concepts of a domain of discourse for the purpose of collaboration and communication. There have been different proposals for the application of ontologies. Niles et al. target the creation of an high-level ontology for defining general-purpose concepts [10]. This ontology can be used as a foundation for other domain specific ontologies. On the other hand, other approaches attempt to design ontologies to describe parts or all of the concepts of a specific domain.

So far, numerous methodologies, methods and tools have been proposed for developing ontologies [4, 3]. In this paper, we are concerned with studying and analyzing ontology development and maintenance frameworks from a domain-centric point of view. By frameworks we mean those proposals that target the main parts of the ontology development process. They specify particular phases for building and managing ontologies and usually provide implemented components for each phase.

Our objective is to study the applicability and suitability of general-purpose ontology frameworks for creating domain-centric ontologies. We intend to analyze the methods for integrating domain specific information into such frameworks. We attempt to investigate how ontologies that are developed by a framework for a specific domain are extracted from domain specific knowledge and information, how they fulfill domain necessities and how they evolve based on arising requirements and changes.

We propose a domain-centric comparison model for analyzing and investigating ontological frameworks. This model has five dimensions. Through the 'Resources' dimension, we study the domain resources which are exploited by a framework in its ontology development process. The 'Scope and usage coverage' dimension specifies the types of applications which can utilize a developed ontology and how the scope of the ontology is determined.'Fineness and independency' examines how much domain granularity an ontology can reach through the employment of the development framework. It also investigates whether a framework can be applied to all domains or it is only designed for a specific one. The 'Maintenance capability' and 'Integration capability' dimensions study how completely a framework supports the maintenance of ontologies and how well it exploits other related ontologies in its process, respectively.

In Section 2, we classify a range of ontological frameworks. We select instances from each category and discuss more about their proposed processes. In Section 3, we fully introduce the proposed comparison model. Section 4 provides some concluding remarks.

2. Ontological Frameworks

We classify ontological frameworks into three main categories. In the first category, we are interested in the frameworks that attempt to take into consideration all of the possible activities of the ontology development process. These

activities include the development of one or more ontologies for a domain of discourse, managing them and letting them to be automatically or manually changed, and keeping and manipulating several versions of the target ontologies. We have selected Text-To-Onto [7] and OntoLearn [9] as two case studies and will analyze them within our proposed domain-centric comparison model.

In the second category, the main aim of the selected frameworks is the management of ontologies, instead of their design. They mostly focus on managing ontologies, integrating several ontologies of the same domain and their evolution rather than constructing ontologies. These types of frameworks can be integrated with other discussed frameworks in order to cover all activities of the ontology engineering process.

Integrating several ontologies related to a specific domain is an important aspect of ontology maintenance. For a domain of discourse, there may be more than one ontologies that have been designed by developers with dissimilar approaches that describe different parts of the same domain. Integrating these ontologies has been the topic of several research papers. Wache et al. [15] give a good survey explaining possible ways for ontology integration. They argue over three approaches of using ontologies for describing a domain of discourse. Single ontology approach stands for a situation that there is only a global ontology for a domain. In this environment there may be other particular ontologies, but all of them should be mapped to the general one. Multiple ontologies, is the second approach discussed in [15]. In this approach there are several ontologies in the system that may be developed independently. In order to integrate such diverse ontologies, inter-ontology mapping algorithms are necessary. PROMPT [11] is a framework that has the appropriate capabilities for maintaining ontologies based on this approach. It has components that provide inter-ontology mappings, merging and alignment. In Section 2.3 we will talk about the PROMPT framework more completely. The third approach is the hybrid ontology approach, where there are several ontologies for a domain, but all of them share a common vocabulary (or upper ontology).

The third category of the selected frameworks is devoted to the frameworks which aim at developing or maintaining ontologies related to a very specific domain. We have chosen the medicine domain, one of the most popular domains that ontologies have been employed. So far, many ontologies as well as methodologies and methods have been proposed in bio-medicine, molecular biology and generally speaking, the medicine domain. Open Biomedical Ontology (OBO) library [2] is a collection of bio-ontologies which have been designed and shared across medical and biological domains. The Gene Ontology (GO) project provides 'a controlled vocabulary to describe gene and gene product attributes in any organism' [1]. We have selected Oasis [12], an integration framework for biomedical ontologies, which best fulfils our definition of ontological frameworks and will study it thoroughly in Section 2.4.

2.1 Text-To-Onto

Maedche et al. [7] have developed an almost mature ontology learning framework. The development process of this framework consists of the following steps:

Importing and reusing other ontologies: In this step, domain experts identify and select appropriate ontologies, schemas and conceptualizations which are related to the domain of discourse. This framework suggests making use of the FCA-MERGE [13], a bottom-up merging method, for integrating domain ontologies.

Ontology Extraction: This step is the major part of the proposed framework. The framework includes OntoEdit as a workbench to help experts collaboratively design ontologies; however, the framework also considers learning ontologies from resources other than experts. Text-To-Onto is a method suggested to be used for extracting ontology primitives from natural language documents. It includes several algorithms for extracting each ontological element so that experts or ontology engineers can choose the appropriate one under different conditions. The Text-To-Onto method first extracts domain related concepts by processing natural language documents using a shallow text processor. Domain concepts are selected based on frequency measures like TF-IDF. In the second step, the method hierarchically clusters domain concepts to form a taxonomic relationship between them. Non-taxonomic relations are the other ontological primitives that are extracted in the third step. Text-To-Onto applies association rule mining algorithms in order to acquire the relationships between the concepts.

Pruning the result ontology and refinement: The result ontology from previous phases may focus on some unnecessary details but may not include some important and useful information. In the pruning phase, some parts of the ontology are removed and then in the refinement phase some ontological elements are inserted. These steps are the ontology management parts of the framework.

This framework has an evolving nature. After each round, ontology engineers can decide to begin the process cycle again. In each phase, the information extracted in the previous cycles can be utilized as background knowledge.

2.2 OntoLearn

Navigli et al. [9] have designed a framework for learning domain based ontologies from relevant documents. Their ontology engineering framework is comprised of three phases: First, the main domain concepts and taxonomic relationships between them are automatically extracted from

domain documents. In the next phase, these concepts are evaluated by the experts through a groupware package called ConSys, and finally all the new accepted concepts are inserted into the ontology using the SymOntoX tool [8]. SymOntoX accepts new concepts and relations from ConSys as incoming suggestions and inspects and applies them to the target ontology.

Phase one of the suggested framework has been realized by the OntoLearn system. OntoLearn itself has three phases. First, terminology concepts are extracted using a linguistic and syntactic parser. The extracted terminologies in the first phase may be compound phrases which consist of several terms. For example, the system may recognize 'bus service', 'coach service' and 'ferry service' as domain terminologies. Every term in a complex terminology may have several senses. Identifying the best sense for each term in the compound phrase is a challenging task. The second phase of OntoLearn is comprised of three sub-phases. First with a novel algorithm named SSI and using upper taxonomic knowledge sources like WordNet, one sense is assigned to each term in a complex phrase. In the second subphase, the OntoLearn method determines appropriate hierarchical relationships between compound terms by clustering synonym and similar terms as well as taxonomic information of WordNet. In the third sub-phase, for extracting non-taxonomic relationships between the components of a compound phrase, an inductive learner is employed. First, an ontology engineer manually tags some instances of domain terms with appropriate relationships relevant to the domain, and then the learner builds a tagging model. The inductive learning system needs a vector of features for each instance. The authors in [9] suggest that the feature vector of each compound phrase should be shaped by all of the hyperonyms of its terms. The OntoLearn framework includes methods for creating and managing ontologies in an evolving manner. Each new concept or relation is learnt by OntoLearn and validated by the experts and inserted through SymOntoX. The management tasks are done in a collaborative manner. SymOntoX define three levels for manipulating ontologies. A simple user can only view the ontology, a super user can read and write, and an ontology master can modify the ontology and validate the suggestions of the super users.

2.3 PROMPT

PROMPT [11] is an ontology management framework which is developed at the Stanford University. The main aim of PROMPT is to provide features for multiple ontology management tasks which include the maintenance of ontology libraries, import and reuse of other related ontologies, support for ontology versioning , merging, mapping and aligning ontologies and factoring independent sections of the ontology. PROMPT consists of four components:

iPROMPT is a component which helps its users merge different ontologies describing the same domain of discourse. Given two ontologies, iPROMPT finds similar concepts between them and suggests them as merging candidates to the ontology engineers. The users can follow the suggestions or select other concepts for merging.

AnchorPROMPT finds similarity between different ontologies; therefore, it facilitates their mapping and alignment. It contains a graph-based algorithm which finds similar ontological concepts of two ontologies, based on a given set of identified anchors.

PROMPTDiff controls ontology versioning. Utilizing some defined heuristics, this component attempts to match ontological elements of two versions of the same ontology and finds the changes that have been applied in the new version compared to the last one.

PROMPTFactor is a component that allows its users factor some part of a huge domain ontology for a specific use.

2.4 Oasis

Oasis [12] is an integration framework for bio-medical ontologies. It provides a warehouse for bio-medical concepts, saves them, finds similarity between the concepts of different ontologies and interactively with the domain experts, maps similar concepts onto each other. The Oasis framework consist of two major parts: a database and a novel mapping tools named IOMG. The Oasis framework has been mainly influenced by the OBO ontologies structure. The database is comprised of three types of tables. Firstly, the OBO tables which include terms and term2terms. Term2term are all ontological terms that have an is-a or has-a relationship with each other. Secondly, Mapping tables are tables that store the concepts that have been mapped by the framework. Finally, proprietary tables that maintain concepts from other ontologies rather than OBO.

IOMG is a tool designed for finding similarity between domain terms and mapping them. It also utilizes a graph-based representation of the OBO ontologies for finding possible similarities. Oasis defines three similarity metrics: linguistic similarity between term names, similarity between the definition of ontological terms that are provided for most of GO terms and also similarity between parents and children of each term in the graph representation. Based on these metrics, IOMG finds a similarity value between each pair of concepts for two given ontologies. IOMG attempts

to find proper pairs from all of the candidate pairs by maximizing an objective function of all similarity values. Consequently, not only similarity of two terms is important for applying a mapping, but also the best conditions under which all of the mapped pairs have their highest similarity values is also considered. Ultimately all established links that cause a cycle in the graph are identified and introduced to domain experts as potential failure links.

3. A Domain-centric Comparison Model

The major goal of our proposed model is that given a specific domain for examination, it would assist the analysts in the selection process of an appropriate framework from the already existing huge number of methods and approaches in order to best fulfil the target domain's necessities. In the following, we thoroughly explain each of its dimensions and analyze the introduced frameworks based on these dimensions.

3.1 Domain Resources

Different frameworks suggest dissimilar information resources for extracting ontological elements. In many methods and frameworks, domain experts are reliable references for the development of ontologies. Based on this fact, the experts' role may significantly vary: some times the only references for every thing are the domain experts who solely or collaboratively design and maintain ontologies, while in other approaches experts only confirm what has been extracted by machine learning algorithms. For example in the PROMPT framework, the only available domain resources are the domain experts who modify ontologies whereas in the OntoLearn Framework, experts confirm and finalize ontological concepts extracted from other domain resources. In another approach, such as the model proposed in the Text-To-Onto framework, domain experts may specify information resources that should be utilized by automatic learning algorithms.

The upper ontology is another learning reference which may be used in the ontology development process. For instance,OntoLearn uses WordNet for disambiguating terms in compound phrases and extracting taxonomic and non-taxonomic relationships between terms. The other domain resource for developing ontologies are semi-structured or natural language Web documents that may be exploited by automatic algorithms for extracting concepts and relationships. Database schemas, domain thesauri and dictionaries are examples of semi-structured information resources. The Text-To-Onto and OntoLearn frameworks make use of domain related Web documents for extracting terminologies and relationships. Furthermore, the former one also exploits semi-structured domain resources. Oasis uses the synonym table of OBO along with the definitions related to each GO concept for finding similarity between medical concepts.

3.2 Domain Scope and Usage Coverage

Specifying the scope of the target ontology is an important subject which has been discussed in various methodologies. The first phase of Uschold and Grninger's methodology [14] is to specify the scope of the ontology. The specification phase of METHOONTOLOGY [5] also aims to find the scope of the ontology. What is important for us in this comparison model is how different frameworks specify the scope of the target ontology. Domain experts have an important responsibility in most cases. The domain scope of a framework can be investigated through experts and all of the available information. While there have been no suggestions for specifying the scope of an ontology in some frameworks, some others utilize a combination of these methods.

Observably, experts are the only reference for defining the ontology scope in frameworks which their development process is completely based on human efforts. They should decide on the degree of relevancy of a new concept to the domain and whether it should be inserted into to the ontology or not. Alternatively, some frameworks attempt to develop or evolve domain ontologies based on all of the available information. The OntoLearn framework aims to extract all of the ontological concepts and relationships from Web documents. However, experts can also influence identifying the borders of the developed ontology by confirming or rejecting ontological elements through the ConSys component. The Text-To-Onto framework also follows a similar approach.

In usage coverage, we consider the types of applications that can benefit from the designed ontology. We show whether all existing applications can utilize the created ontology or only specific ones which has been indicated by the domain experts can benefit from it. Navigli et al.[9] suppose that the coverage of an ontology should be very vast to make it useful. They state that all of domain related concepts should be available in a suitable ontology. Based on such definition, all of the applications related to that domain should be able to make use of the designed ontology and hence communicate on this basis. On the other hand, some frameworks attempt to evolve the domain ontology based on the users' needs. In this approach, only domain applications which are specified by the domain experts and ontology engineers can exploit the designed ontology. The PROMPT and Oasis frameworks are examples of such an approach.

3.3 Maintenance Capabilities

Several frameworks have been suggested for developing and managing ontologies. As was explained previously, some of the frameworks emphasize more on constructing ontologies whereas the others focus more on maintaining and evolving them. However, the ontology learning and constructing frameworks that we have chosen provide partial capabilities for ontology evolution.

The maintenance capabilities criterion investigates how thorough a framework supports the maintenance of domain ontologies. We have studied two items in this field: evolution support and also the quality of ontology manipulation. Preserving different versions of an evolving ontology, undoing changes applied to a version and retrieving the latest versions of an ontology are considered as factors indicating the degree of evolution support. Among the introduced ontological frameworks, only PROMPT provides evolution support to some extent. The PROMPTdiff component of the PROMPT framework is designed for representing changes between different versions of the ontology without making use of change logs. PROMPT considers conditions where the selected ontologies have been developed in different environments so a unified change log is not accessible. The framework does not mention other versioning activities like reversing the applied changes by particular users.

Ontology manipulation can be realized through either expert collaboration or the evolving lifecycle of the development process of a framework. The most popular approach in ontology maintenance is to allow domain experts manipulate ontologies and change it according to their needs. Nonetheless, implementation of this simple schema in a distributed environment could have some practical difficulties. Through our comparison model, we examine whether a framework provides the experts with the possibility of collaboratively managing ontologies in a distributed environment or not. The Text-To-Onto and OntoLearn frameworks, are instances of frameworks that allow experts manage ontologies in a collaborative manner.

An evolving lifecycle is another way to adapt a designed ontology to new conditions. New changes and modification can be learnt in each cycle during ontology development. Both OntoLearn and Text-To-Onto exploit this type of lifecycle for updating an under-development ontology over time.

3.4 Integration Capabilities

For a specific domain, there may exist other ontologies that have already been designed. Ontology development and management frameworks might utilize these ontologies in their process. In the proposed comparison model, we want to investigate how a framework exploits other existing ontologies. Some frameworks provide no integration capability, such as the case in the OntoLearn framework which does not suggest any method for utilizing other ontologies. As another approach, a framework may make use of other related ontologies for developing one inclusive domain ontology, such as the Text-To-Onto framework. Some other frameworks like Oasis and PROMPT follow another approach for integration. They attempt to concurrently integrate several ontologies for a domain of discourse in a 'multi-ontology environment'.

3.5 Domain Fineness and Independency

Domain fineness is concerned with how much granularity a framework attempts to achieve while enriching the domain ontology. Similar linguistic words may have different meanings in different domains. A good framework should be able to deal with each word based on its specific semantic in the target domain. Furthermore, domain specific relationships might exist that can only be extracted when a framework exploits domain knowledge and heuristics. The OntoLearn framework exploits domain specific information and knowledge for extracting ontology concepts to some extent. The disambiguation algorithm assigns domain related sense to each term of a compound phrase. Furthermore, in the case of non-hierarchal relationships among classes, the framework makes use of domain relative information for generating rules which tag compound terms with appropriate non-taxonomic relationships. The Oasis framework also utilizes domain specific features for integrating ontologies. OBO graph representation as well as the definitions that are provided for GO concepts is used by Oasis to find similarity between the concepts.

Through domain independency, we study dependency or independency of a framework to a specific domain. One can investigate whether and with how much accuracy a framework can be applied to different domains. In some circumstances, there is a tradeoff between these two criteria. The frameworks that attempt to apply too many domain specific heuristics and knowledge might be not completely domain independent. Among the introduced frameworks, Oasis is the only one which is completely dependent to the medical domain and cannot be employed by other domains.

4. Concluding Remarks

Ontological frameworks are those that attempt to suggest ontology engineers a process for developing or maintaining ontologies. They attempt to clarify each step of the process by providing an implemented component. They have not been designed for end-users nor applications. They target engineers who want to develop ontologies for the end-users. We have designed a model for comparing and investigating

Table 1. Analysis of the Ontological Frameworks From Domain-centric Perspective.

	Text-To-Onto	OntoLearn	PROMPT	Oasis
Experts' Role	Selection of Other Resources, Approve and finalize ontological elements learnt by the method.	Approve and finalize ontological elements learnt by the method.	Responsible for all tasks	Responsible for all tasks
Web Docs	Natural language documents from the Web	Domain-based web documents	-	-
Upper Ontologies	-	WordNet	-	-
Semi-Structured Resources	domain thesaurus and domain related schemas	-	-	OBO synonym tables and the definition related to each GO concept
Model for Scope Specification	All information/Domain Experts	All information/Domain experts	Domain experts	Domain experts
Usage Coverage	All applications	All applications	Specific applications	Specific Application
Evolution Support	-	-	Versioning	-
Ontology Manipulation	Evolving development life cycle&Collaborative manip.	Evolving development life cycle&Collaborative manip.	Standalone manip.	Standalone manip.
Integration Capabilities	Based on global ontology	-	Multi-ontology environment	Multi-ontology Environment
Domain Fineness	-	Use of domain knowledge for concept extraction	-	Use of domain knowledge for Integration
Domain Independency	Independent	Independent	Independent	Doamin dependant

such frameworks. Our framework focuses more on the domain capabilities of each framework. We investigate how an ontological framework can utilize domain knowledge and also how it can be used for a certain domain of interest. We extracted five criteria for investigating such frameworks. While some of criteria focus on generating a general view of the application of a given framework, the others attempt to find domain relevancy of it and also domain closeness of the generated results. Table 1 summarizes the results of the performed comparisons.

Considering the domain perspective, our main attempt for future research is to design a high-level methodology which assists engineers in developing domain-centric ontological frameworks. This methodology would allow the designers to specify the goal of the framework and its capabilities, the domain knowledge it intends to exploit, the domain coverage of the final ontology which is designed by the framework and also its dependency or independency to a specific domain. This methodology would be a meta-methodology which is designed not for developing ontologies but for crafting frameworks which manage and design ontologies.

References

[1] The gene ontology, http://www.geneontology.org/, last visited feb 2007.

[2] Open biomedical ontologies, http://obo.sourceforge.net, last visited feb 2007.

[3] O. Corcho, M. Fernandez-Lopez, and A. Gomez-Perez. Methodologies, tools and languages for building ontologies: where is their meeting point? *Data Knowl. Eng.*, 46(1):41–64, 2003.

[4] Y. Ding and S. Foo. Ontology research and development. part 1- a review of ontology generation. *Journal of Information Science*, 28(2):123–136, 2002.

[5] M. Fernandez-Lopez, A. Gomez-Perez, J. P. Sierra, and A. P. Sierra. Building a chemical ontology using methontology and the ontology design environment. *IEEE Intelligent Systems*, 14(1):37–46, 1999.

[6] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, 43(5-6):907–928, 1995.

[7] A. Maedche and S. Staab. Ontology learning for the semantic web. *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]*, 16(2):72–79, 2001.

[8] M. Missikoff and F. Taglino. Symontox: a web-ontology tool for ebusiness domains. pages 343–346, 2003.

[9] R. Navigli and P. Velardi. Learning domain ontologies from document warehouses and dedicated web sites. *Comput. Linguist.*, 30(2):151–179, 2004.

[10] I. Niles and A. Pease. Towards a standard upper ontology. In *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, pages 2–9, New York, NY, USA, 2001. ACM Press.

[11] N. F. Noy and M. A. Musen. The prompt suite: interactive tools for ontology merging and mapping. *Int. J. Hum.-Comput. Stud.*, 59(6):983–1024, 2003.

[12] G. Song, Y. Qian, Y. Liu, and K. Zhang. Oasis: A mapping and integration framework for biomedical ontologies. In *CBMS '06: Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*, pages 611–616, 2006.

[13] G. Stumme and A. Maedche. FCA-MERGE: Bottom-up merging of ontologies. In *IJCAI*, pages 225–234, 2001.

[14] M. Uschold and M. Gruninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11:93–136, 1996.

[15] H. Wache, T. Vogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hubner. Ontology-based integration of information - a survey of existing approaches. In *workshop on Ontologies and Information Sharing at the International Joint*, pages 108–117, 2001.