INTRODUCING ONTOLOGY EVOLUTION MANAGEMENT

Marcelo Paulo Amaolo

Departamento de Ciencias de la Computación, Universidad Nacional del Comahue

Buenos Aires 1400, Neuquén, Argentina

Phone: (54) 299 - 4490312, fax: (54) 299 4490313

E-mail: mamaolo@uncoma.edu.ar

Abstract:

Recently, the interest in the use of ontologies —which can be seen as an explicit, formal specification of a shared conceptualization of a domain of interest— has increased, as they are becoming an integral part of many industrial and academic applications. We describe the Ontology Evolution Process as a framework to understand the problems that emanate from management of ontologies evolution. We present some considerations about the compatibility among evolving ontologies and related approaches.

1 INTRODUCTION

In recent years ontologies have become a topic of interest in computer science. The term Ontology in its original sense is a philosophical discipline, a branch of philosophy that deals with the nature and the organization of being [Smi99].

There are different definitions in the literature of what an ontology should be, some of them are discussed in [Gru02, Gru93, Gua98, Smi99, Sur03]. Gruber [Gru93] gives the following definition:

> *"An ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an ontology is a systematic account of Existence. For Artificial Intelligence systems, what "exists" is that which can be represented".*

A *conceptualization* refers to an abstract model of some phenomenon in the world by identifying the relevant concept of that phenomenon. *Explicit* means that the types of concepts used and the constraints on their use area explicitly defined.

This definition is often extended in the computer field [Sur03] by three additional conditions: "An ontology is an explicit, *formal* specification of a *shared* conceptualization *of a domain of interest*". In this case, *formal* refers to the fact that an ontology should be machine readable, excluding natural languages. *Shared* reflects that an ontology captures consensual knowledge, accepted normally by a group of individuals. And the reference to *a domain of interest* indicates that domain ontologies aren't be used in modeling the whole world, but just parts which are relevant to the task at hand.

Ontologies are a key technology in any project that needs formal description of particular domains. Today, ontologies are becoming an integral part of many industrial and academic applications in the fields such as supporting semantics-based search, interoperability support, configuration support, constraint specification and validation, Semantic Web applications, and others [Noy03]

The current excitement about the vision of a Semantic Web forms an additional stimulant for the interest in ontologies. In this vision, ontologies have a role in defining and relating concepts that are used to describe data on the web. The use of ontologies on the web emphasizes particular research topics [Gru02, Hef00, Kle01, Kle02b, Mae04, Sun02]. Reuse of ontologies will be a central aspect because of the interlinked nature of the web. In the idea of a Semantic Web, relative small ontologies link to many other ontologies to import vocabularies and domain theories. The distributed and dynamic character of the web also emphasizes an issue that was not yet extensively studied: the evolution and versioning of ontologies [Kle02b, Mae04, Noy03].

However, applying those pieces of knowledge to the real world problems has some critical points, such as the fast changes on the domain and user needs. Ontologies will inevitably change over time, and support to handle this evolution is needed. This is specially important when ontologies will be used in a decentralized and uncontrolled environment like the web, where changes occur without coordination [Sun02]. This may have unexpected and unknown results, like incompatibilities in the applications and ontologies that refer to them, and could give wrong interpretations to data o even make data inaccessible [Kle03, Mae04].

In this paper we introduce a description of the Ontology Evolution Process [Mae02b] as a framework to understand the problems that emanate from management of ontologies evolution and the considerations that must be observed to comply with the different kinds of compatibilities that this problem introduces.

The rest of the paper is organized as follows. In the next section, we start with causes that can initiate changes in ontologies. In section 3, we describe and explain the main phases of the ontology evolution process. After that, we discuss in section 3, considerations of compatibilities among changing ontologies, and we conclude the paper in section 5.

2 Ontology Changes

There are several reason for changes in ontologies. Starting from the proposed definition [Kle02c, Kle03, Mae02a, Ma02b], changes in ontologies can be caused by either:

– ***Changes in the domain***: several situations in which changes in the real world (domain evolution) require changes that the ontology describing this domain also must take. For example consider the division of two university departments: this real world change must be attended with changes on the ontology that describes this modified domain.

– ***Changes in the conceptualization***: the conceptualization of the domain is not a static specification that is produced once, but has to be reached over time. This construction is achieved with agreement in a social process of exchanging information and meaning, where users perspectives result in different views on the domain and as a result in a different conceptualization. For example, consider the merging of two university's departments and the changes that arise with the lending priorities in the new merged library.

– ***Changes in the specification***: a change in the way in which a conceptualization is formally recorded. The translation from one knowledge representation language to another generates changes in the explicit specification, as languages probably differ not only in their syntax but also in their semantics and expressivities.

This variety of causes and consequences of the ontology changes makes ontology evolution a very complex process and should be considered as both an *organizational* and *technical* problem.

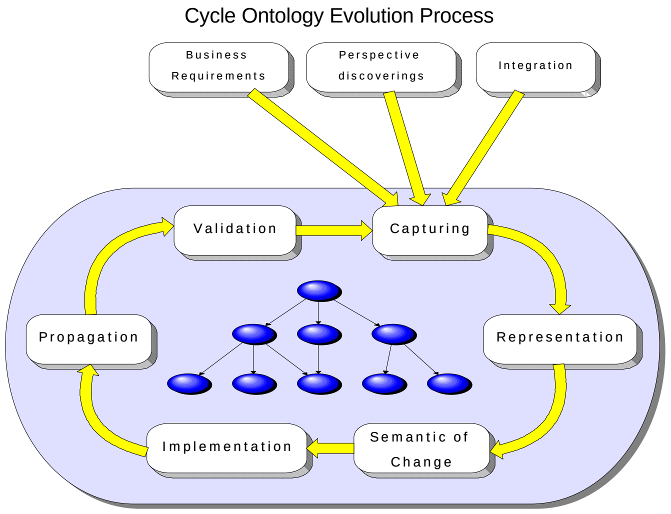

Figure 1: Cycle Ontology Evolution Process

3 Ontology Evolution Process

Ontology evolution [Sto03] can be defined as the timely adaptation of the ontology to the changed business requirements, to the trends in the ontological instances and to the way of using of the ontology-based applications. Ontology Evolution is also the consistent management / propagation of these changes because a modification in one part of the ontology may generate subtle inconsistencies in other parts of the same ontology, in the ontology-based instances, depending ontologies and applications.

The overall process has a cyclic structure and is described in Figure 1. We describe shortly each of the phases in the following.

* Change representation:

To settle changes, the modifications must be identified and represented in a suitable format. Changes can be specified in a fine grain level, when elementary changes in the ontology are required. Or they can be specified in a more thick level, introducing composite changes representing a collection of elementary changes applied together. This specification eludes the successive application of a list of elementary evolution changes that could produce unnecessary states if each change is applied alone. This mechanism permits generation of more semantic, that could be lost in a compose application of elementary changes [Kle02b, Kle03, Noy02].

* **Semantic of change:**

Application of an elementary change in the ontology can induce syntax and semantic inconsistencies in other parts of the ontology. Syntax inconsistencies arise when undefined entities at the ontology or instance level are used or ontology model restrictions are invalidated. Semantic inconsistencies arise when the meaning of an ontology entity is altered.

The task of semantics of change phase is to facilitate resolution of induced changes in a systematic manner, ensuring consistency of the whole ontology. Thus, to resolve a change, the evolution process needs to establish answer at many resolution points (branch points during change resolution) where taking a different path will generate different results. Each possible answer at each resolution point is an elementary *evolution strategy*. An evolution strategy [Sun02] unambiguously defines the route how elementary changes will be resolved. Typically a evolution strategy is chosen by the user at the start of the ontology evolution process and it is circumscribed to organization outlines.

* **Change implementation:**

In order to prevent from performing undesired changes, before applying a change to the ontology, a list of all implications to the ontology should be generated and presented to the user, as he should be able to understand the list and approve or cancel the modification.

* **Change propagation:**

All dependent artifacts related to the modified ontology must be replaced with the latest versions. First, ontology instances need to be changed to preserve consistency with the new ontology, on a broad process to reach them all. Second, as ontology reuse and extend other ontologies, an ontology update could distort other ontologies that depend on it, in addition to all the artifacts that are based on these ontologies. A recursive use of the ontology evolution process could solve this problem. However, special care must be taken to consider, besides syntax inconsistencies, the semantic inconsistencies that arise when the ontology includes a concept that is already added in the original ontology. Third, the ontology evolution approach must identify and take into account all applications based on the modified ontology whose functionality is affected, and take steps correspondingly.

It was also proposed two ways [Mae02b] for the changing deployment:

- **Push-based approach**: Changes from the changed ontology are propagated to dependent ontologies as they *happen*. Suitable when strict dependent ontology consistency is required, since the information about the original state of the changed ontology is available for the evolution of the dependent ontology.
- **Pull-based approach**: Changes from the changed ontology are propagated to dependent ontologies only at their *explicit request*. This approach is better suited for less stringent consistency requirements, as ontologies may be temporally inconsistent and the recovering of the consistence of dependent ontology is more difficult.

Other implications must be considered to practice ontology development in distributed environments like the web [Hef00].

* **Validation:**

In order to enable recovering from circumstances when ontology engineers take unsatisfactory decisions and approve changes that shouldn't be performed, this phase is introduced as experimental purpose and enables validations of performed changes and undoing at user's request. The reversibility means canceling all effects of some change, which may be not simply inverse change manually. Creating evolution logs typically solves the problem of reversibility. An evolution log, based on the evolution ontology described, tracks information about each change, allowing recreation of the sequence of changes leading to current state of the ontology.

In [Kle03] is discussed how the information about change can be represented in different ways and describe some techniques and heuristics that supplement information could be missing or not considered.

* **Change discovery and capture:**

We can differentiate two types of changes:

- **top-down**: explicit changes, regularly demanded by top-manager looking for adjusting the system to new business requirements. It can be realized by any system with limited ontology evolution tools [Sun02, Mae02b, Sto02a].

- **bottom up**: implicit changes, discovered only by the analysis of the system's behavior or the resulted structure. For example, a taxonomy group containing no instances for a long period of time could be a candidate for removing. As another example an ontology subconcept that, after several deletes of sibling subconcepts of the same concept from different sources, finished with only one subclass, a strike at the original purpose of the classification, that could be conceptualized together with its class in the new ontology.

4 COMPATIBILITY FACTOR

In order to determine which changes to an ontology are backward compatible, we need to determine what compatibility means [Kle03, Mae04]. We can distinguish, among others, some dimensions that we must consider when determining whether a new version of an ontology is compatible with the old one:

- **Instance data preservation:** no data is lost in transformation from the old version to the new one (instances)
- **Ontology preservation:** a query result obtained using the new version is a superset of the result of the same query obtained using the old version
- **Consequence preservation:** if an ontology is treated as a set of axioms, all the facts that could be inferred from the old version can still be inferred from the new version
- **Consistency preservation:** if an ontology is treated as a set of axioms, the new version of the ontology does not introduce logical inconsistencies.

When we characterize the effects of the changes operations we need to take these, and possible other dimensions into account.

It was also proposed [Mae02b] that the backbone of the whole evolution process is a meta-ontology for evolution that enables representation, analysis, realization and sharing ontological changes in a more systematic and consistent way. This ontology can be used to provide basement for the development of tools and mechanisms to derive new pieces of information from existing information.

This ontology can be build considering basic change operations [Mae02b] with an extension that defines complex change operation. Basic operations form a set rich enough to specify the changes between the old and the new ontology and to capture knowledge about the change to derive new information. Complex change operations are composed operations of multiple basic operations that incorporate some additional knowledge about the change. The logical entity implied by the complex operations enables determine the effects of operations more precisely.

Therefore, this meta ontology must be build based on an algebra of ontology operations which must include these composite operations in order to analyze the predictability of the changes as completely different class than each of the simple operations that constitute it.

5 CONCLUSIONS AND NEXT STEPS

The management of changes is the key issue in the support for evolving ontologies. In contrast to the ontology evolution that allow access to the ontology itself and all dependent artifacts, through the newest ontology, ontology versioning allows access to data through different versions of the ontology. Thus, ontology evolution can be treated as a part of the ontology versioning. The trouble of finding differences between versions of ontologies is in fact very closed to the problem of ontology merging. While in the merge the process is finding the similarities, in the evolution we can treat the process as complementary, finding the differences.

Several alternatives have been proposed to attack the problem of change in ontologies [Mae04, Sto02a, Sun02], but ontology evolution as a whole process has not been completely treated, as the research in ontology evolution is in its very early stage [Noy03]. The study has been focus as part of the development of ontology editors and ontology environments [Sto02a]

We discussed the fact that ontologies are not static, but evolve over time. Domain changes, adaptations to different tasks, or changes in the conceptualization require modifications of the ontology. We describe the elements of the ontology evolution process and the considerations that must be considered to respect compatibility between old and new versions of ontologies.

We believe that can be interesting to investigate the requirements that must be accomplished by editors and environments to manage the evolution process. In our next steps we are going to identify the means and mechanism used in the current developments and compare their analysis and achievements. Another aspect to consider is the implications related to the ontology evolution process in organization and what mechanisms and tools must be develop to fulfill the process in a distributed environment like the web, in order to support the evolution of the ontologies with its consistency kept.

6 REFERENCES

[Gru02] Gruninger, Michael, Lee, Jintae, *"Ontology applications and design"*, Comm. Of the ACM, February 2002.

[Gru93] Gruber, T.R., *"A translation approach to portable ontology specification"*, Knowledge Acquisition, 1993.

[Gua98] Guarino, Nicola, *"Formal Ontology and Information Systems"*, Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy, 1st edition

[Hef00] Heflin, Jeff and Hendler, James, *"Dynamic Ontologies on the Web"*, In Proc. 7th National Conference on Artificial Intelligence AAAI-2000. AAAI/MIT Press, 2000.

[Hel03] Hellman, Ziv and Gal, Amit, *"Structural conflict avoidance in Collaborative ontology Engineering"* in Proc. Fifth International Conference on Enterprise Information Systems, Angers, France April 23 - 26, 2003

[Kle01] Klein, Michel and Fensel, Dieter, *"Ontology versioning on the Semantic Web"*, In Proceedings of the First International Semantic Web Working Symposium (SWWS), pages 75-91, Stanford University, California, USA.

[Kle02a] Klein, M., Fensel, D., Kiryakov, A., and Ognyanov, D., *"OntoView: comparing and versioning ontologies"*, In Collected Posters ISWC 2002, Sardinia, Italy.

[Kle02b] Klein, Michel *"Supporting evolving ontologies on the Internet"*, In Lindner, W. and Stuller, J., editors, Proceedings of the EDBT 2002 PhD Workshop, number 2490 in LNCS, pages 597-606, Prague, Czech Republic.

[Kle02c] Klein, Michel; Kiryakov, Atanas; Ognyanov, Damyan and Dieter Fensel, *"Finding and characterizing changes in ontologies"*, In Proc. 21st International Conference on Conceptual Modeling (ER2002), number 2503 in LNCS, pages 79-89, Tampere, Finland.

[Kle03] Klein, Michel and Noy, Natalya F., *"Component-Based Framework For Ontology Evolution"*, In Proc. Workshop on Ontologies and Distributed Systems, IJCAI '03, Acapulco, Mexico. Also available as Technical Report IR-504, Vrije Universiteit Amsterdam.

[Mae02a] Maedche, A., Motik, B., Stojanovic, L., Studer, R. and Volz, R. *"An Infrastructure for Searching, Reusing and Evolving Distributed Ontologies"*, in Proc. International World Wide Web Conference 12th International Conference on World Wide Web, Budapest, Hungary

[Mae02b] Maedche, A., Motik, B., Stojanovic, L., Studer, R. and Volz, R. *"Managing Multiple Ontologies and Ontology Evolution in Ontologging"*, In Proc. Conference on Intelligent Information Processing, World Computer Congress 2002, Montreal, Canada. Kluwer Academic Publishers, 2002.

[Mae04] Maedche, Alexander and Staab, Steffen, Chapter 5: Ontology management - Storing, aligning and maintaining ontologies. *Handbook on Ontologies 2004*, S. Staab, R. Studer (eds.). Springer Verlag, 2004.

[Noy02] Noy, Natalya F. and Klein, Michel, *"Ontology Evolution, not the same as schema evolution"*, Technical Report SMI-2002-0926, Stanford Medical Informatics.

[Noy03] Noy, Natalya F. and Musen, Mark, *"Ontology Versioning as an Element of an Ontology-Management Framework"*, Technical Report SMI-2003-0961, Stanford Medical Informatics.

[Smi99] Smith, Barry, *"Ontology and Information Systems"*, from Ontology: Philosophical and Computational (draft), *http://ontology.buffalo.edu/smith/articles/ontologies.htm*

[Sto02a] Stojanovic, Ljiljana and Motik, B., *"Ontology Evolution within Ontology Editors"*, Proc. OntoWeb-SIG3 Workshop, 13th International Conference on Knowledge Engineering and Knowledge Management EKAW 2002, Siguenza (Spain), 30th September 2002

[Sto02b] Stojanovic, Ljiljana; Stojanovic, Nenad and Handschuh, Siegfried, *"Evolution of the Metadata in the Ontology-based Knowledge Management Systems"*, In Proc. 1st German Workshop on Experience Management: Sharing Experiences about the Sharing of Experience, Berlin, March 7-8, 2002.

[Sun02] Sunagawa, Eiichi; Kozaki, Kouji; Kitamura, Yoshinobu and Mizoguchi, Riichiro, *"An Environment for Distributed Ontology Development Based on Dependency Management"*, The Semantic Web - Dieter Fensel, Katia P. Sycara, John Mylopoulos (Eds.): Springer 2003

[Sur03] York Sure, *"Methology, tools and case studies for ontology based knowledge management"*, PH Tesis, University of Karlsruhe, May 2003.