HOW TO EVOLVE ONTOLOGY AND MAINTAIN ITS COHERENCE

A Corrective Operations-based Approach

Najla Sassi, Wassim Jaziri and Faiez Gargouri
Higher Institute of Informatics and Multimedia, Sfax University, Tunisia

Keywords: Ontology Evolution, Corrective Operations, Evolution Kits.

Abstract: An ontology is a specification of a conceptualization related to a domain of knowledge. In an evolution context, it cannot be considered as a finite conceptualization since it must be adapted to new requirements. This adaptation must respect the coherence of the ontology and its conformity regarding some objectives. To update ontology while maintaining its coherence, this paper proposes an anticipatory approach based on corrective operations. For each change to occur on the ontology, we define corrective operations to prevent and correct potential inconsistencies likely to be generated.

1 INTRODUCTION

Changing environments require adaptable ontologies to changes that occur over time. The adaptation of ontology is a complex process and several evolution problems must be treated, in particular maintaining of the ontology coherence. The ontology coherence is an agreement between its ontological entities with respect of semantics of the subjacent language of representation.

The application of a change in ontological entities is a modification of a subset of knowledge represented by the ontology. Changes management requires defining mechanisms specifying how knowledge can be changed and how to maintain the consistency of knowledge after each change. In addition, ontological entities are linked to each other semantically (their semantics are complementary) and conceptually, the application of a change in some ontological entities may have effects on other entities. Thus, the ontology evolution requires a structured process to take into account all direct and indirect effects of any change.

Two types of inconsistency may be identified (Maedche et al., 2003):

- Structural inconsistency occurs when the constraints of the ontology model are invalid or if the semantics of the subjacent language of ontology is not respected.
- Semantic inconsistency occurs when the significance of the entities of ontology is changed.

Maintaining the ontology coherence remains little studied in the literature. In (Maedche et al., 2003), ontology is considered consistent if its axioms are respected and if it satisfies the whole of the invariants defined in the model of ontology. The authors defined constraints of consistency related to the model of ontology according to the semantics of the KAON language.

In (Schlobach et al., 2003), an algorithm of resolution of inconsistencies is proposed based on the identification and the elimination of incoherent concepts. This algorithm identifies the concepts sources of "logic contradictions" and provides intelligent algorithms to follow and solve the sources of inconsistencies.

Stojanovic et al. (Stojanovic et al., 2003) proposed an approach for the management of evolution and the maintaining of consistency for KAON ontologies. The authors proposed the concept of strategies of evolution which allow to the ontologist to choose the most suitable solutions for the resolution of inconsistencies.

Haase et al. (Haase et al., 2005) also used the concept of strategies of resolution based on the constraints of OWL-Lite for the detection and the resolution of inconsistencies in OWL ontologies. However, the resolution of inconsistencies is done after application of changes. The resolution of

Sassi N., Jaziri W. and Gargouri F. (2009).
HOW TO EVOLVE ONTOLOGY AND MAINTAIN ITS COHERENCE - A Corrective Operations-based Approach.
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, pages 384-387

inconsistencies is ensured in two phases: the detection of inconsistencies which consists in finding the parts of ontology which do not satisfy the consistency conditions and the generation of changes that allow ensuring the consistency of ontology by generating additional changes. Three types of consistency are defined: structural consistency, logic consistency and user consistency. Structural consistency ensures that ontology satisfies the constraints of ontology language. Logic consistency refers to the formal semantics of ontology and at its satisfiability, i.e., it is semantically correct and does not present logic contradictions. The user consistency takes into account the particular requirements of users.

Flouris et al. (Flouris et al., 2005) differentiate between a consistent ontology and a coherent ontology. Ontology is inconsistent if there is no interpretation which satisfies all the axioms of this ontology. It is incoherent if it does not satisfy some predefined constraints or the related invariants. The predefined constraints describe the consistent model of ontology. These authors consider the inconsistencies as sign of bad design and their correction does not relate to the ontology evolution but it is rather related to the ontology design.

Luong et al. (Luong et al., 2007) distinguish two levels of consistency for the model of ontology: structural consistency and logic consistency. Structural consistency relates to the constraints of consistency defined for an ontology model by ensuring a good organization of the ontological entities at the level of structure. Logic consistency checks if the elements of ontology remained "semantically correct" after their evolution. The inconsistencies generated in ontology can be solved automatically using strategies of resolution. These strategies contain solutions which guide the process of resolution for all the types of changes.

However, the majority of existing works are interested in specific categories of ontology such as OWL ontologies or KAON ontologies. Moreover, the proposed approaches are based on the correction of inconsistencies after they occur. In this paper, we propose an anticipatory approach to manage inconsistencies before they occur. We express the requirements of evolution using types of changes. For each type of change, we define corrective operations that must be applied in conjunction with this type of change in order to correct consistencies.

This paper is structured as follows. Section 2 proposes an anticipatory approach for ontology consistency management. We present in sections 3 and 4 the principles of corrective operations and evolution kits. Sections 5 concludes this work.

2 ONTOLOGY EVOLUTION APPROACH

We propose in this work an ontology evolution approach based on three steps to allow monitoring the evolution of ontology by creating a new version better adapted to the required changes (figure 1):

1. *Expressing evolution changes*: in changing environment, users express new requirements to take into account in the ontology. These requirements are expressed informally and sometimes in a fuzzy and ambiguous manner. In this step, we express clearly the users' requirements according to types of changes to apply on the ontology.
2. *Maintaining the ontology coherence*: each type of change may generate inconsistencies in all parts of the ontology. In this step, we verify the effects of changes on the ontology and we define corrective operations to resolve them. In addition, knowledge represented by the ontological entities is complementary and dependant, it is also necessary to identify the direct and indirect effects (the derived effects) of each type of change.
3. *Creating a new version*: after updating ontology by applying types of changes, a new ontology version is created. Thus, in an evolution context, different versions of ontology should coexist. To control these versions, it is important to monitor the relationship between them. However, establishing links between versions is a complex task and requires an investment. These links must respect the order of versions and the changes have been occurred [Kle02] [NK04]. We also decide on the relevance to preserve the old version of ontology in the ontological database or to remove it. This choice is conditioned by the types of implemented changes (subtractive or not subtractive changes). In the case of a subtractive evolution, the old version of the ontology will be stored and added to the ontological database. It is also important to provide access to all versions of ontology.

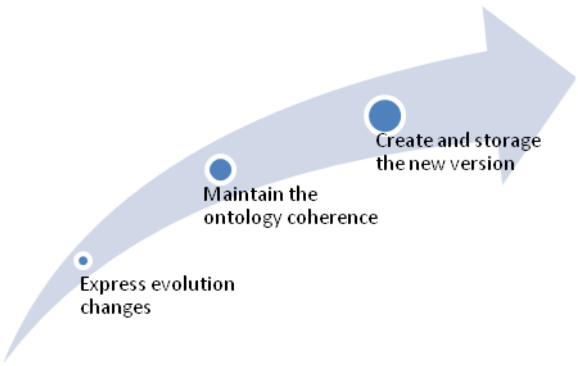

Figure 1: The steps of an ontology evolution process.

3 CORRECTIVE OPERATIONS

The evolution of ontology can be expressed by the update of one or more ontological entities (concept, relationship, property, axiom). To allow updating an ontological entity, we define primitive and composite operators called types of changes able to evolve ontology. These types of changes extend these proposed in the literature (Klein et al., 2002) (Stojanovic, 2004) to express all evolution possibilities on the ontological entities (Sassi et al., 2008). Each requirement of evolution can be expressed by a primitive or a composite type of changes.

In an evolution process, the application of types of changes should have as consequence an ontology which is in conformity with the whole of coherence rules. The preservation of the ontology coherence requires the preservation of the integrity of the model and the constraints of ontology by preventing the effect of each type of change on the ontology. However, types of changes ensure only the modification of ontology. They not guarantee that ontology remains coherent after modifications. The definition of types of changes must be associated with adequate mechanisms to ensure the coherence of ontology and its conformity after evolution. This task is essential in an ontology evolution process since it conditions the validation and the adoption of the new generated version of ontology.In this work, we develop anticipatory solutions managing the inconsistencies upstream of their appearance to avoid them. We identify the inconsistencies due to each type of change in order to propose corrective operations changes allowing correcting them. These corrective operations are automatically applied by the system in combination with the type of change.

To define corrective operations, we analyse the direct and indirect effects of each type of change, we detect inconsistencies likely to be generated on the ontological entities and define additional changes for each type of inconsistencies to resolve them (figure 2). Corrective operations depend on the type of change.

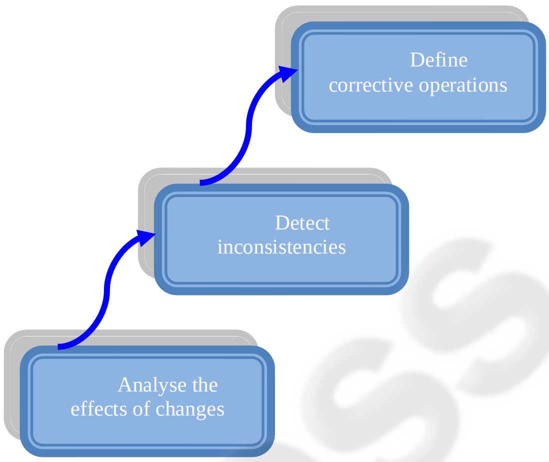

Figure 2: The definition of corrective operations.

4 EVOLUTION KITS

We define an evolution kit as the combination of a type of change and the corrective operations. The evolution kits allow updating ontology while preserving its coherence. We define for each evolution kits: the type of change, the pre-conditions, post-conditions, potential inconsistencies and additional changes.

- Pre-conditions: must be checked and controlled by the system before applying a type of change.
- Inconsistencies: potential problems can be generated due to a type of change.
- Additional changes: to be attached to each type of change to correct the inconsistencies that may be generated.
- Applicative post-conditions: define what must be true after applying the type of change, independently of the ontology coherence.
- Coherence post-conditions: define what must be true if the ontology is coherent.

Each type of change represents with additional changes, a "coherent evolution kit". We define as many evolution kits as types of changes (Sassi et al., 2008). For an evolution requirement, the corresponding coherent change kit is applied rather than only the type of change.

5 CONCLUSIONS

This paper treats the problem of ontology evolution and the coherence maintaining. It presents corrective operations to allow updating ontology while maintaining its coherence and its conformity. Types of change allow updating ontology but do not ensure its coherence. The application of a type of change may produce inconsistencies in ontological entities. To correct them, corrective operations are automatically done in addition to the type of changes.

To implement evolution kits, we developed the OntoChanges tool based on Protege. OntoChanges is an ontology evolution support which allows users updating ontologies while preserving theirs coherences.

REFERENCES

Flouris, G., Plexousakis, D., 2005, Handling Ontology Change: Survey and Proposal for a Future Research Direction. Technical report FORTH-ICS/TR-362.

Haase, P., Stojanovic, L., 2005, Consistent Evolution of OWL Ontologies. In Proceedings of the 2nd European Semantic Web Conference (ESWC-05).

Klein M., Fensel D., Kiryakov A., Ognyanov D., 2002, Ontology versioning and change detection on the web, 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02), Spain.

Luong, P-H. Dieng-Kuntz, R. 2007, A Rule-based Approach for Semantic Annotation Evolution, The Computational Intelligence Journal, 23(3):320-338, USA.

Maedche, A., Motik, B., Stojanovic, L., 2003, Managing Multiple and Distributed Ontologies in the Semantic Web, VLDB Journal - Special Issue on Semantic Web, 12, 286-302.

Sassi, N., Jaziri, W., Gargouri, F., 2008, Formalisation of evolution Changes to update domain ontologies, In Proceedings of International Arab Conference on Information Technology (ACIT'2008), Hammamet, Tunisia.

Sassi N., Jaziri W., Gargouri F., 2009, Using of ontology for modeling the evolution of Tunisian higher education system, 15th conference on Information and Software Technologies Kaunas, Lithuania, 2009.

Schlobach, S., Cornet, R., 2003, Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies, In Proceedings of IJCAI'03.

Stojanovic L., 2004, Methods and Tools for Ontology Evolution, PhD thesis, University of Karlsruhe.

Stojanovic N, Hartmann J., Gonzalez J., 2003, Ontomanager - a system for usage-based ontology management, Proceedings of FGMLWorkshop.