Z-BASED FORMALIZATION OF KITS OF CHANGES TO MAINTAIN ONTOLOGY CONSISTENCY

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

Keywords: Changing Environment, Ontology, Coherence, Kits of Change, Z language.

Abstract: In changing environments, supporting ontology's evolution is essential to integrate changes and to manage ontology versions. It is also important to guarantee the consistency of ontology when changes occur. In this paper, we present an ontology evolution approach based on kits of changes. These kits are based on changes operators and additional changes which correct inconsistencies caused by the changes operators. A formalization of the kits of changes is also proposed based on the Z language.

1 INTRODUCTION

Ontology is an explicit representation of knowledge related to a domain of study and a particular context. The application of changes in its conceptual entities is a modification of a subset of knowledge represented by the ontology. The application of changes requires defining the mechanisms specifying how knowledge can be changed and how to maintain the consistency of knowledge after each change.

Ontology evolution is the process of adaptation of ontology to evolution changes and the consistent management of these changes to guarantee the consistency of ontology when changes occur (Klein et al., 2001) (Noy et al., 2004). It encompasses the set of activities, both technical and managerial, which ensures that ontology continues to meet organizational objectives and users needs in an efficient and effective way (Stojanovic, 2004).

The adaptation of ontology to evolution changes is a complex process from which several problems must be managed: identification of evolution changes, analysis of effects of changes, management of the ontology consistency, storage of ontology versions, etc. We are especially interested in this paper at defining kits of changes to update, in a coherent way, the ontology to new evolution requirements.

This paper is organized as follows: Section 2 presents the evolution approach based on kits of changes. In section 3, we specify the kits of changes using pre-conditions, post-conditions, potential inconsistencies and additional changes. Section 4 defines a formalization of the ontology structure based on the ontology meta-model. The formal specification of kits of changes using Z language is presented in section 5 before concluding in section 6.

2 KITS OF CHANGES

In changing contexts, the management of changes and the maintaining of the ontology consistency require analyzing and identifying effects of changes on all ontology elements as well as defining additional operations to correct inconsistencies.

In our approach, we express the requirements of ontology evolution using types of changes. Indeed, the evolution of ontology is the update of one or more ontological entities. 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: concepts, relationships, properties and axioms (Sassi et al., 2008).

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 consistency

Sassi N., Jaziri W. and Gargouri F. (2009).
Z-BASED FORMALIZATION OF KITS OF CHANGES TO MAINTAIN ONTOLOGY CONSISTENCY.
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, pages 388-391

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 version of ontology. In this work, we develop anticipatory solutions managing the inconsistencies upstream of their appearance to avoid them. We propose a preventive approach to anticipate inconsistencies due to each type of change and to propose additional changes allowing correcting these inconsistencies. These additional changes are automatically applied by the system in combination with the type of change.
Additional changes depend on the type of change. To define additional changes, we identify the types of change likely to generate inconsistencies, analyze these inconsistencies and define additional changes able to correct them. A kit is composed of the type of change and the additional changes.

3 SPECIFICATION OF KITS OF CHANGES

The kits of changes allow updating ontology while preserving its consistency. We define for each kit of changes: 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 kit of change". We define as many kits of changes as types of changes identified in the taxonomy of the types of changes (Sassi and al, 2007).

In addition, we define rules of consistency which ontology must verify to be considered as consistent.

Definition: *a type of change preserves the consistency of ontology if it preserves the rules of consistency.*

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.

- Examples of rules of consistency:

- Define for each domain the key concepts which should not be removed from the ontology: ***Rule (1)***.
- Ontology should not have isolated concepts: ***Rule (2)***.
- A concept must comprise at least a property: ***Rule (3)***.
- Ontology should not contain semantically contradictory information: ***Rule (4)***.
- The semantics of information should not be reversed between ontology versions: ***Rule (5)***.
- An ontological entity must conserve all elements of definition: ***Rule (6)***.
- Ontology should not contain redundancies of data: ***Rule (7)***.

Some rules of consistency, such as Rule (1), are taken into account in the pre-conditions. For example, to remove a concept, it should not be a key concept.

4 FORMALIZATION OF THE ONTOLOGY STRUCTURE

Formalizing the changes semantics requires a definition of the ontology model as well as its change operations. Formalization is based on:

- *Formal methods:* based on mathematics, can be used in any step of the cycle of life of ontology in order to make precise a development process.
- *A language of formal specifications:* used for an abstract representation of ontology.

In this work, we use the Z language and *Z-eves tool* to formally specify the ontology structure and the kits of changes.
Ontology is represented as *a schema.* It is defined as the set of concepts and relationships between them.

$\cup\_Ontology$ _____
$\rightarrow C: \Pi\ Concept$
$\rightarrow RAS: \Pi\ Association$
$\rightarrow RA: \Pi\ Aggregation$
$\rightarrow RC: \Pi\ Composition$

→*RH:* Π *Hierarchy*
→*RS:* Π *Semantic_Relationship*
→*R:* Π *Relationship*
→*A:* Π *Axiom*
→*AR:* Π *Association_Axiom*
→*AC:* Π *Concept_Axiom*
→*key_C:* Π *Key_Concept*
→*key_Sem:* Π *Key_Semantic_Relationship*
→*L:* Π *Partial_Link*
∠_____

A concept is an ontology element, characterized with a noun and a set of properties.

∪_*Concept*_____
→*name: WORD*
→*P:* Φ *Property_C*
∩_____
→*P* ⌊ 0
∠_____

A relationship is characterized with a noun and its related concepts.

∪_*Relationship*_____
→*name: WORD*
→*c1: Concept*
→*c2: Concept*
∠_____

We distinguish conceptual relationships (association, aggregation, composition, hierarchy) and semantic relationships. An association has a noun, a set of related concepts, cardinalities and a set of properties.

∪_*Association*_____
→*Relationship*
→*P:* Π *Property_R*
→*CL:* Π *Concept*
→*AC:* Π *Cardinality_Axiom*
∠_____

To take into account the n-ary associations (n>2), we define the notion of Partial_Link which allows extending an existing association with a link to another concept.

∪_*Partial_Link* _____
→*concept: Concept*
→*rel: Association*
→*AC: Cardinality_Axiom*
∠_____

5 FORMAL SPECIFICATION OF KITS OF CHANGES

We present in this section a formal specification of some kits of changes.

1. Add_Concept:*Syntax*: Add_Concept (cnew)

Z statement:

→*cname?: WORD*
→*c?: Concept*
→*pname?: WORD*
→*rname?: WORD*
→*cnew!: Concept*
→*rnew!: Relationship*
→*pnew!: Property_C*

Pre-conditions: A*c: C* ∞ *cname?* ⌊ *c . name*

Potential inconsistencies:

Rule (2) ; Rule (3)

Applicative post-conditions: C' = C Y {*cnew!*}

Coherence post-conditions:

R' = R Y {*rnew!*} ;

cnew! . P = cnew! . P Y {*pnew!*}

Additional changes:

Add_Relationship (cnew, c, rname)
Add_Concept_Property (c, pname)

2. Rename_Concept:

Syntax: Rename_Concept (c, cnameNew)

Z statement:

→*c?: Concept*
→*cname?: WORD*

Pre-conditions: c? ε *C ;* A*x: C* ∞ *x . name* ⌊ *cname?*
Applicative post-conditions: c?. name = cname?

Coherence post-conditions: Ø

Additional changes: Ø

3. Remove_Concept:*Syntax:* Remove_Concept (c)

Z statement:

→*c?: Concept*

Pre-conditions:

c? ε *C*

Potential inconsistencies:

Rule (6)

Applicative post-conditions: C' = C \ {*c?*}

Coherence post-conditions:

c? . P = 0
A*r: R | r . C1 = c? ϖ r . C2 = c? ∞ R' = R \ {r}*
A*r: RC | r . C1 = c? ∞ C' = C \ {r. C2}*
A*a: AC | a .C1 = c? ϖ a .C2 = c? ∞ AC' =AC \{a}*

Additional changes:

Remove_Concept_Property (c, p)

Remove_Relationship (r, c1, c2)

Remove_Concept_Axiom (c1, c2, a)

Remove_Concept (co)

4. Remove_Association:

Syntax: Remove_Association (r, c1, c2)

Z statement:

→r?: Association

Pre-conditions: r? ε RAS

Potential inconsistencies:

Rule (2) ; Rule (6)

Applicative post-conditions: RAS' = RAS \ {r?}

Coherence post-conditions:

r? . P = 0
A*c: C ∞* E *r: R ∞ r . C1 = c ϖ r . C2 = c*
A*a: AR | a . R1 = r? ϖ a . R2 = r? ∞ AR' = AR \ {a}*
r? . AC = 0

Additional changes:

Remove_Association_Property (r, p)
Remove_Cardinality_Axiom (r, c1, c2).
Remove_Association_Axiom (r, r1, a)

Remove_Concept (c)

Add_Relationship (c, c_o, rname)

5. Remove_Concept_Axiom:

Syntax: Remove_Concept_Axiom (c1, c2, exp)

Z statement:

→a?: Concept_Axiom

Pre-conditions: a? ε AC

Applicative post-conditions: AC' = AC \ {a?}

Coherence post-conditions: Ø

Additional changes: Ø

6 CONCLUSIONS

This paper presented kits of changes to allow updating ontology while maintaining its consistency. An evolution kit is a sequence of a type change and additional changes. The type of change allows updating ontology but does not ensure its coherence. The application of a type of change may produce inconsistencies in ontological entities. To correct them, additional changes are automatically done in combination with the type of changes.
However, other kits of changes can be defined to facilitate the expression of evolution requirements, such as add ontology. Add ontology relates to the problem of ontology merging and require mapping algorithms to compare ontological entities. We will consider this problem in future works.

REFERENCES

Klein M., Fensel, D., 2001, Ontology versioning for the Semantic Web, *International Semantic Web Working Symposium*, USA.

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.

Noy N., Klein, M., 2004, Ontology evolution: Not the same as schema evolution, *Knowledge and Information Systems*, 6(4):428-440.

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. Types de changements et leurs effets sur l'évolution de l'ontologie. *in Proceedings of the Journées Francophones sur les ontologies (JFO'2007).* p.75-93. Sousse. Tunisia. 2007.

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