AN ALGORITHM FOR BUILDING INFORMATION SYSTEM'S ONTOLOGIES

Mohamed Mhiri, Sana Chabaane, Achraf Mtibaa, Faïez Gargouri
MIRACL Laboratory, ISIM Institute, BP 1030-3018, Sfax. Tunisia

Keywords: Ontology, Information systems design, concept, semantic relationships, conceptual relationships.

Abstract: Current applications' modelling becomes increasingly complex. Indeed, it requires a hard work to study the particular studied field in order to determine its main concepts and their relationships. The conceptual representations (CR) results of the modelling of such applications can contain structural and semantic errors which are not detectable by current CASE. The solution that we propose is to associate an ontology, for the studied field, as a help to the designers during IS modelling steps. Building such information system's ontologies require the use of an approach allowing the determination of the concepts and the relationships between these concepts. Using ontologies makes it possible to ensure conceptual representations' semantic coherence for a given field. In this paper, we propose an algorithm for building an information system's ontology based on the comparison between the concepts and using a set of semantic relationships.

1 INTRODUCTION

Information systems are increasingly complex in particular with the enormous growth of the volume of data, the extraordinary evolution of the technologies and the evolutionary requirements of the users. Consequently, current applications require an enormous effort of design and development. In fact, such applications require a detailed study of their fields in order to define their concepts and to determine the concepts' relationships. On the other hand, applications become more and more cooperatives using data from heterogeneous sources, in particular in the case of the Web applications (e-learning, e-commerce, semantic Web, etc). The designers are also confronted with different conflits. A typology of these conflicts is given in (Ouksel, 1999).

In the literature, several approaches were proposed to solve the previously enumerated problems in order to assist the designers in their tasks of modelling, mainly to represent knowledge during the process of design.

Recently, as an evolution and generalization of those approaches, several authors proposed the ontology concept. Ontology is now considered as the most complete and adequate way for resolving the major modelling conflicts. Indeed, *an ontology is an explicit specification of a conceptualization* (Gruber, 1993).

In this paper, we present an algorithm for building information system's (IS) ontologies. It is organised as follows. In the second section, we present our approach for building information system's ontologies and the different semantic relationships used. The third section presents our framework and the corresponding algorithm for building IS's ontologies. We conclude this paper by giving our future works.

2 OUR APPROACH FOR BUILDING IS'S ONTOLOGIES

Our ontology building is based on the three steps proposed by (Leclère, 2002) and which we adapt to IS (Mhiri[1], 2005). The *conceptualization* step consists in identifying the knowledge contained in a particular field. The *ontologization* step consists in modelling, in a generic language, the formal properties of the considered field. The last step, the *operationalization* supports an inference mechanism allowing the reasoning implementation.

An ontology's modelling requires the introduction of new semantic and conceptual relationships which can exist between the concepts.

Mhiri M., Chabaane S., Mtibaa A. and Gargouri F. (2006).
AN ALGORITHM FOR BUILDING INFORMATION SYSTEM'S ONTOLOGIES.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - ISAS*, pages 467-470
DOI: 10.5220/0002449304670470

A lexical study of different domains allows as to identify the following relationships (semantic and conceptual ones) which can exist between the concepts. They are described as follows.

- *Synonymy relationship,* exists between two different concepts which express the same meaning. The mutual implication, i.e. the fact that one of the terms implies the other, and that the second implies the first, can be seen like a proof of synonymy.
- *Equivalence relationship,* a concept can express in a particular context the same meaning with another concept.
- *Identity relationship*, exists between two identical concepts having the same meaning.
- *Homonymy relationship*, the same concept can have two different meanings.
- *Antonymy relationship*, between two concepts indicates that the negation of the first implies the assertion of the other; we can't deny both at the same time.
- *is_a Relationship*, it is a relationship of classification between a general concept and a specific one.
- *Classification relationship*, it is a particular case of the is_a relationship. It can be expressed by the *kind_of* relationship.
- *Aggregation relationship* represents the *set/elements* relationship.
- *Composition relationship:* it is a strong aggregation in which a concept can be composed of one or several other concepts.

In the following section, we present an algorithm of IS's ontologies building.

3 AN ALGORITHM FOR BUILDING IS'S ONTOLOGIES

Our proposed algorithm is based on four steps (Figure 1).

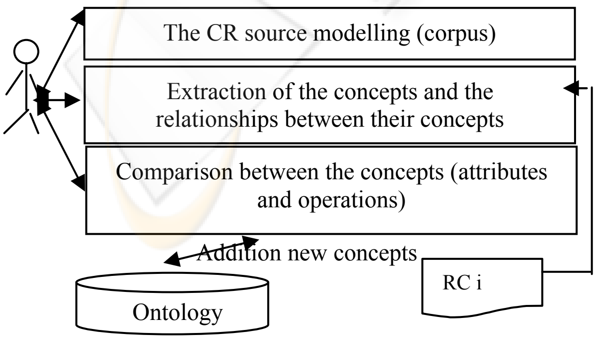

Figure 1: Building steps of IS ontology.

These steps are assisted by the designers allowing the corpus modelling, the checking and the correction of the obtained results.

3.1 Concepts and Their Relationships Extraction

The determination of the concepts and the relationships between them are based on several CR presenting a given field (e-commerce for example).

The extraction of the concepts consists in the extraction of the classes' names, their properties (attributes and operations) and their relationships.

We define a set of rules, presented at the following section, allowing the comparison between the concepts of the different CR.

3.2 Comparison Between the Concepts of the Different CR

The comparison step is based on the concepts' semantic. This semantic is expressed by each concept's attributes and operations. The following algorithm determines the semantic relationships between CR concepts; it is represented by the following rules, where we consider C1 and C2 as concepts. We define the following functions :

Name(C) : the concept's name
Attribute(C) : the set of all C attributes
Operation(C) : the set of all C operations.

Rule 1 :

Identity (C1, C2) $\Leftrightarrow$
Name(C1) = Name(C2) ^
Attribute(C1) = Attribute(C2) ^
Operation(C1) = Operation(C2).

Rule 2 :

Homonymy (C1, C2) $\Leftrightarrow$
Name(C1) = Name(C2) ^
Attribute(C1) $\neq$ Attribute(C2) ^
Operation(C1) $\neq$ Operation(C2).

Rule 3 :

Synonymy (C1, C2) $\Leftrightarrow$
Name(C1) $\neq$ Name(C2) ^
Attribute(C1) = Attribute(C2) ^
Operation(C1) = Operation(C2).

Rule 4 :

Kind_Of(C1, C2) $\Leftrightarrow$
Name(C1) $\neq$ Name(C2) ^
Attribute(C1) χ Attribute(C2) ^
Operation(C1) χ Operation(C2).

Rule 5 :

Equivalence(C1, C2) $\Leftrightarrow$

Name(C1) ≠ Name(C2) ^
Operation(C1) χ Operation(C2).

Rule 6 :

Name(C1) ≠ Name(C2) ^
Attribute(C1) ≠ Attribute(C2) ^
Operation(C1) ≠ Operation(C2)⇔ ambiguity (C1, C2).

We can propose to the designer two solutions:

1. Add the concept introduced like a new entry to our ontology.
2. Choose a semantic relationship in the proposing list of the concepts composing our ontology.

Rule 7 :

Antonymy(C1, C2) ⇔
Name(C1) ≠ Name(C2) ^
Attribute(C1)= ! Attribute(C2) ^
Operation(C1) = ! Operation(C2).

All these rules are used to elaborate the comparison algorithm.

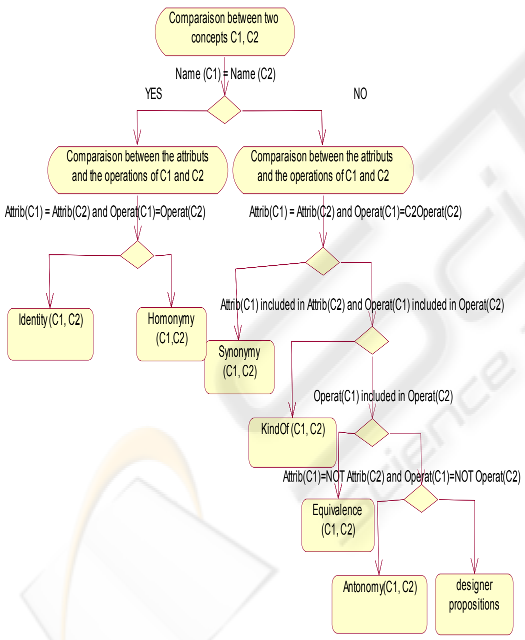

Figure 2: Comparison between the concepts of the different CR.

In the following section, we represent our ontology with an ontology's definition language (ODL).

3.3 UMLOnto Ontology Representation

The representation of our ontology requires the use of a comprehensible and semi-formal language. For that, we proposed a language UMLOnto (Mhiri[2], 2005), based on UML. UMLOnto allows representing the concepts as well as the relationships between these concepts.

The diagram obtained is called concepts diagram. This diagram can be represented by an XML file by adding new tags supporting the characteristics of ontologies. Indeed, XML is a standard language supported by the major current CASE especially using UML.

We used the tag <concept > representing a concept which can be a class or an association. A concept is characterized by its attributes and its operations. The relationships between concepts can be:

-conceptual relationship <association>, <aggregation>, < composition >, < inheritance >,

- semantic relationship which can be synonymy, homonymy, equivalence, kind_of, antonomy.

The name of this ontology is characterized by a tag < name_Ontology >.

We also developed a framework to implement our approach. It is based on ArgoUML[1], which is an open source software using UML (figure. 3).

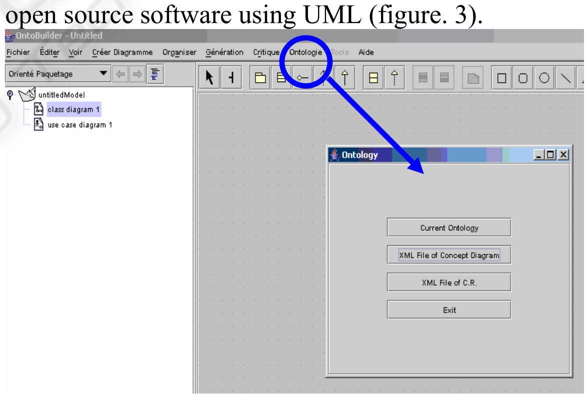

Figure 3: Framework interface.

In the following section, we present the fourth step allowing the update of our ontology.

3.4 Ontology Update

The update of our ontology consists in the iterative using of the algorithm of comparison with the CR provided by the designer with the previous version of ontology.

[1] http://argouml.tigris.org/

The ontology update must preserve its structure. One can have two types of updates:

- The corrective update,
- The evolutionary update.

This step is still in the experimentation in our group.

4 CONCLUSION

In this paper, we presented an algorithm for building IS' ontologies. It allows the extraction of the concepts and the relationships between concepts of different CR. Then, we used an algorithm of comparison between the concepts based on semantic relationships. This comparison makes it possible to determine the relationships between the concepts of the different CR. To represent our ontology, we proposed UMLOnto a language which is comprehensible by the designers. This ontology is expressed with XML by adding new tags allowing the exchange between the CR. Our ontology building approach is progressive and iterative.

Like prospective for this work, we will formalize the concepts diagram with language Z and to check the coherence of our ontology. Then, we will define functionalities of an ontology maintenance and will integrate them in CASE.

REFERENCES

Gruber, T.R., 1993.Toward Principles for the Design of Ontologies, Used for Knowledge Sharing. Stanford Knowledge Systems Laboratory.

Leclère, L., Trichet, T., Frûst, F., 2002. Construction of an ontology related to the projective geometry. *RFIA 13th congrès des Reconnaissance des Frames et Intelligence Artificielle*, France.

Mhiri[1], M., Mtibaa, A., Gargouri, F., 2005. Towards an approach for building information systems'ontologies". *FOMI'2005,1st workshop Formal Ontologies Meet Industry*, Verona, Italy, June, 9-10.

Mhiri[2], M., Mtibaa, A., Gargouri, F., 2005. UMLOnto: Towards a language for the specification of information systems'ontologies. *SEKE'05, the Seventeenth International Conference on Software Engineering and Knowledge Engineering*, Taiwan, China, July, 14-16, p.743-746.

Ouksel, A., Sheth, A., 1999. Semantic Interoperablity in Global Information Systems. ACM SIGMOD Record, March, Vol 28, N°1, p. 5-12.