Model to describe multi-leveled ontologies and their relations

EL HASSAN ABDELWAHED
Département d’informatique, Faculté des Sciences Semlalia Marrakech
BP 23 90, Bd My Abdellah Marrakech MAROC
Tel: 212 44 43 46 49, Fax: 212 44 43 67 69

Abstract: - Within the last years, many ontologies are created and are accessible over the web. On the other hand, ontologies are used in many applications to explicit formalization of the conceptualization of the domain, but a given domain can be conceptualized through many different ontologies. Thus multiples ontologies connected by semantic relations emerge as a core question and became an essential issue of interoperability between distributed applications over the web. In this paper, we present a model and methodology for describing ontologies according to different levels of abstraction. The functional level contains generic ontologies and the domain level contains domain ontologies. The latter are described from the former by using a derivation process. The proposed formalism is based on an algebraic approach and it allows to capture the relations over different described ontologies.

Keywords: Multi-layered ontologies, Algebraic approach, Model, ontologies mapping, Derivation

1 Introduction
In the early 90’s, the ontologies are originally developed in the knowledge engineering community to support the knowledge sharing and reuse. An ontology provides a set of best founded constructs that can be used to build meaningful higher level knowledge. [7] defines an ontology as an explicit specification of conceptualization. In fact, an ontology is the theory about how a given domain is structured, what sorts of concepts it contains, what sorts of relationships exist among them, and so on. With the emergence of the semantic web, the ontology research has been accelerated and advocated as support of interoperability between distributed applications over the web. The main purpose of an ontology is to support exchange and communication, without misunderstanding, among users and agents. Nowadays, ontology is being put to use for many practical purposes in many other areas. For more detailed descriptions and bibliography of the field see the published survey: [1] gives a review of ontology-based approaches to semantic integration, [2] provides a survey of the most relevant ontological modeling approaches and [3, 8] describe the current state of the art of ontological engineering and it covers theory, tools and applications.

1.1 Motivation
Different autonomously developed applications can meaningfully collaborate and communicate by using a shared ontology. Unfortunately, an unique shared ontology describing all possible domains is very fast to develop and unachieved in practice. In addition, the same domain can be described by different ontologies in a heterogeneous way: the very same concept can be described in different manner and at a different level of detail by more than one ontology. On the other hand, within the last years, many ontologies are created and are accessible over the web. Semantic interoperability, therefore, can be solved by relating different ontologies via semantic mapping [4].

Guarino classifies ontologies by considering two criteria: level of detail and level of dependence. Several types of ontologies can be distinguished. In particular, there are the top-level ontologies and the domain ontologies [5]. In [6] the authors propose a multi-layered spatial ontologies definition framework for urban applications.

1.2 Contribution
In this paper, we present formalism for describing the multi-leveled ontologies and their relations. To illustrate this formalism, we consider an example from the urban domain (see Figure 1). In the proposed formalism, different ontologies are described according different levels of abstraction. The functional level contains generic ontologies and the domain level contains domain ontologies. The derivation process permits to describe ontologies of the domain level from those of the functional level. This aspect of defining ontologies by level allows to have minimal ontologies by level. It avoids defining a global and maximal ontology which is difficult to specify completely for a given domain. The proposed formalism is based on an algebraic approach. Entities of the functional level are represented by...
abstract classes. Each class is described by its intension which is constituted by a set of its attributes. The attributes take values in abstract classes or in abstract data types (ADT). We use the algebraic specification to describe the ADT[13,14]. Entities of the domain level are represented by the concrete classes. The latter are described from the abstract classes by derivation process. This process consists mainly in interpreting abstract data types by domains of values and each abstract class by set of the concrete classes (carriers). We illustrate the proposed formalism by examples from the urban domain (see Figure 1).

Fig. 1. Example: Multi-leveled urban ontologies

1.3. Organization of the paper
The remainder of the paper is organized as follows. Section 2 discusses background and related issues of ontologies and ontology mapping. It present the components of our approach and the methodology used to describe ontologies according multiple levels. The proposed formalisms is detailed in sections 3 and 4 and it is illustrated by examples from the urban domain. Section 5 studies the relations between ontologies. Section 6 presents some related work. Finally, section 7 concludes the paper.

2 Multi-leveled ontologies
2.1 Motivated example
To illustrate our methodology and model to describe ontologies according multiple levels, we consider an example from the urban domain. This methodology is based on notions of abstraction and organization of the information in thematic layers within the spatial systems. Each thematic layer is characterized by a generic set of functionalities. For instance, the entities of the thematic layers (roads networks, electric networks, water pipe networks, etc.) share common functional features. So they can be described by a generic functional entity (called network) based on graph terminology and graph traversal operations which includes: inter-nodes distance evaluation, path cost optimization, path traversal, etc. This description is made without any implication of application domain. For example, the nature of objects that traverse the graph is less important at this level of abstraction. This information will be clarified at the lower level where these objects will be cars in the case of the road networks or the flows in the case of the drinking water system. Similarly, a two-dimensional spatial object (called coverage) can be used to define the generic functional features of the entities of the thematic layers (Cadastre, land occupation, buildings, parks, etc.) including: surface computation, distance evaluation, adjacency test, etc.

2.2 The proposed approach
To meet the above design requirements, each layer is organized into two levels (see Figure 1 and Figure 2):
- Functional level: it contain generic entities including: Node, Link, Surface, etc. ; each entity is mainly represented by an abstract functional description in the generic ontologies.
- Domain level: it contains specific entities to a given domain (roads networks, Cadastre, land occupation, electric networks, etc.). Their descriptions are obtained from the generic entities by a derivation process. Those descriptions constitute the domain ontologies.

[9, 10] adopt an algebraic approach to described ontologies and their relation. In this case, the ontologies are presented as logical theories: an ontology is defined as a pair \( O=(S,A) \) where \( S \) is the ontological signature describing the vocabulary and \( A \) is a set of ontological axioms specifying the intended interpretation of the vocabulary in some domain of interest. The relations between ontologies are described by means of ontology morphisms. For more detailed descriptions of the field of ontology mapping and Semantic Integration technologies, see the published survey [11,12].

As we will see in the next sections, the proposed model in this paper is based on the such algebraic approach. In this model, the ontologies are organized in many layers (see illustration from urban application, Figure 2). Each
layer contains the entities that share common functional features and are described according to two levels of abstraction. In the functional level, generic ontologies $O_G$ describe entities using abstract classes $\xi$. In the domain level domain ontologies $O_D$ use concrete classes $\Psi$. Our ontologies specify entities with attributes to describe their proprieties and function describing their functional aspect.

3 Functional level

3.1 Introduction

Formally, $O_G$ is described by a set of abstract classes $\xi$ described with their intensions. The classes of $\xi$ are organized in hierarchy by the Isa (⊆) and Part-of relations. Each generic entity $e_G \in O_G$ is defined by abstract class $\alpha \in \xi$. The proprieties $e_G$ are described by the attributes of $\alpha$ and the functions of $\alpha$ define the behavior of $e_G$. Constraints expressed by axioms can be used to describe semantics of attributes and functions of the abstract class $\alpha$.

3.2 Formalization

Let’s designate by:
- $T$ a set of abstract data types (ADT) defined by an algebraic specification $(S, \Sigma, E)$ where $S$ is a many ordered-sorted.
- $P_{\alpha}$ a set of attributes of $\alpha \in \xi$ and by $P$ the set of all attributes: $P = \cup_{\alpha \in \xi} P_{\alpha}$

Note that, $\forall \alpha \in \xi, \forall \beta \in \xi, \alpha \leq \beta \Rightarrow (P_{\beta} \subseteq P_{\alpha})$

Let’s designate by $Typep$, an application that associate the attributes of classes of $\xi$ with their types:

$$Typep : \alpha \in \xi \times P \to \xi \cup T$$

$\forall \alpha \in \xi, \forall p \in P_{\alpha}$, two cases are possible:
- $Typep (\alpha, p) = s \in T$: the attribute $p$ take values in ADT specified by a sort $s$. Note that $Typep (\alpha, p)$ can be a basic type (integer, real, string, etc.)
- $Typep (\alpha, p) = \lambda \in \xi$, here the attribute $p$ describe a relation (Part-of) between $\alpha$ and $\lambda$.

Let’s designate by $CardMin$ and $CardMax$ applications that define the minimum and maximum cardinality constraints of the attribute $p$ of $\alpha \in \xi$:

$$CardMin : \xi \times P \to N$$

$CardMax : \xi \times P \to N$, $N$ is a set of natural numbers (integer).

Formally, generic ontology is described, to a high level of abstraction, by a hierarchy of abstract classes $\xi$. Each class $\alpha \in \xi$ is described its intension. This latter contains a set $P_{\alpha}$ of attributes and a set $FA_{\alpha}$ of functions. Each function $f \in FA_{\alpha}$ is represented by its profile defining the types of its arguments. Those latter and attributes types are defined by ADTs or by abstract classes.

**Definition 1 (Intension $(I_{\varphi_{\alpha}})$):** The intension of abstract class $\alpha \in \xi$, noted $(I_{\varphi_{\alpha}})$, is given by:

$$(I_{\varphi_{\alpha}}) = \langle P_{\alpha}; Typep_{\alpha}; CardMin_{\alpha}; CardMax_{\alpha} \rangle$$

such as $\forall p \in P_{\alpha}$:

$$Typep_{\alpha} (p) = Typep (\alpha, p);$$

$$CardMin_{\alpha} (p) = CardMin (\alpha, p);$$

$$CardMax_{\alpha} (p) = CardMax (\alpha, p);$$

Note that: $\forall \alpha \in \xi, \forall \beta \in \xi, \alpha \leq \beta \Rightarrow (I_{\varphi_{\beta}} \subseteq I_{\varphi_{\alpha}})$

**Definition 2 (Generic ontology):** Generic ontology $O_G$ is defined by:

$$O_G = \langle T; \xi; Isa_{\xi}; Typep_{\xi}; CardMin_{\xi}; CardMax_{\xi} \rangle$$

- $T$: Set of ADTs described by algebraic specification $(S, \Sigma, E)$ where $S$ is a many order-sorted.
- $\xi$: Hierarchy of abstract classes described in intension by $I_{\xi}$
- $I_{\xi}$: Set of the intensions of abstract classes, $I_{\xi} = \cup_{\alpha \in \xi} (I_{\varphi_{\alpha}})$ where $(I_{\varphi_{\alpha}})$ is the intension of $\alpha \in \xi$

3.3 Example

For example, consider the set of generic spatial entities: \{Position, Node, Link, Oriented-Link\}, described as follow:

**Node**

position : Position

isolated : Boolean

Get-position(N : Node) → P : Position

Is-Isolated(N : Node) → B : Boolean

etc.

**Position**

x : Number

y : Number

Get-x (P : Position) → X : Number

Get-y (P : Position) → Y : Number

Put-x (X : Number) → P : Position

Put-y (Y : Number) → P : Position

Identique (P1 : Position, P2 : Position) → I : Boolean

Identique (p1, p2) \Rightarrow

(Get-x (p1) = Get-x (p2)) \& (Get-y (p1) = (Get-y (p2))

**Link**

n1, n2 : Node

Oriented : Boolean

Weight : Number

Create-Link(N1 : Node, N2 : Node) → A : Link

Get-Weight(A : Link) → P : Number

etc.

**Oriented-Link ≤ Link**

orientation : Orientation

Get-orientation(A : Oriented-Link) → O : Orientation

etc.
According to the definition 2 above, we have:
\[ \xi = \{ \text{Position, Node, Link, Oriented-Link, ...etc.} \} , \]
\[ T = \{ \text{Number, Boolean, ...etc.} \} , \]
\[ P_{\text{Link}} = \{ n1, n2, \text{Oriented, Weight} \} , \]
\[ \text{Type}_p (\text{Link, n1}) = \text{Node} \]
\[ \text{Type}_p (\text{Link, Weight}) = \text{Number} , \]

4. Domain level

4.1 Introduction

The domain level contains entities that are specific to a given domain. Those entities are organized in the thematic layer. For instance, the entities of water networks, roads networks, electric networks, etc. are regrouped in the same layer. For a given layer, we use domain ontology \( O_D \) to describe the entities of each thematic. Formally, the domain ontologies \( O_D \) are represented by a set of concrete classes \( \Psi \). Those classes are described by their intensions and interpreted by their extensions. The classes of \( \Psi \) are organized en hierarchy.

4.2 Formalization

Let’s designate by :

- \( \Psi \) a set of the concrete classes. Note by \( \Psi_x \) the set of classes of \( \Psi \) that are directly derived from classes of \( \xi \) (\( \Psi_x \subset \Psi \)).
- \( \Omega \) the set of instance of \( c \in \Psi \) and by \( \Omega \) the set of all instances, \( \Omega = \bigcup_{c \in \Psi} \Omega_c \).
- \( \Phi \) an application that associate each class \( c \in \Psi \) with its instances: \( \Phi (c) = \Omega_c \) such that
  \[ \forall b \in \Psi \ \forall c \in \Psi : b \leq c \Rightarrow \Phi (b) \subseteq \Phi (c) \]
- \( \Gamma_\xi \) the application defined from \( \xi \) to \( \Psi \). This application associates an abstract class \( \alpha \in \xi \) with a concrete class \( a \in \Psi \).

The class \( a, \Gamma_\xi (\alpha) = a \), represents the interpretation of \( d \alpha \) or its derivative.

- \( \alpha^d \subset \Psi_x \) the set of a concrete class that directly derived from \( \alpha \in \xi \).
  \[ \alpha^d = \bigcup_{\Gamma_\xi (\alpha)} \text{with } \alpha^d \subset \Psi_x \]

For instance, if \( \alpha = \text{Link} \) then \( \alpha^d = \{ \text{Road-Link, Electric-Link, etc.} \} \) where \( \text{Road-Link} \) and \( \text{Electric-Link} \) are classes of \( \Psi \). Note that, \( \forall c \in \alpha^d, P_c = P_o \).

That is the class \( c=\Gamma_\xi (\alpha) \) inherit the attributes and functions of \( \alpha \in \xi \). In this case: \( P = \bigcup_{c \in \alpha^d} P_c \).

Similarly, we have, \( \forall c \in \alpha^d, \forall p \in P_c : \)
\[ \text{CardMin}(c, p) = \text{CardMin}(\alpha, p) ; \]
\[ \text{CardMax}(c, p) = \text{CardMax}(\alpha, p) \]
Let’s designate by \( \Gamma_T \) the application that associate each sort \( s \in S \) with its interpreting domain \( D_s \ :
\[ \Gamma_T : S \rightarrow M_S \]

\( M_S \) is the model of algebraic specification \( (S, \Sigma, E) \) that is used to describe \( T \) in the functional level.

Let’s designate by \( \text{Dom} \) an application that associate an attribute \( p \in P \) for \( c \in \Psi \) with its values domain noted \( \text{Dom}(c, p) \) :
\[ \Psi \times P \rightarrow D_c \] where \( D = \Omega \cup M_d \) such that
\[ \forall c \in \Psi, \forall p \in P_c : \]
\[ \text{Dom}(c, p) = D_c \] if \( \text{Type}_p (\alpha, p) = s \in S \), with \( D_c = \Gamma_T (s) \).

Note that, \( \forall c \in \Psi, \forall d \in \Psi' : \)
\[ c \leq d \Rightarrow (\text{Dom}_d \subseteq \text{Dom}_c) \]

Definition 3 (Intension \( I_{\Psi} \)) : The intension of a concrete class \( c \in \Psi \), noted \( I_{\Psi,c} \) is given by:
\[ I_{\Psi,c} = < P_c, \text{CardMin}_c, \text{CardMax}_c > \]

where \( \forall p \in P_c, \forall f \in F_c, \forall \alpha \in \Pi : \)
\[ \text{Dom}(p, f) = \text{Dom}(c, p), \text{CardMin}(p) = \text{CardMin}(\alpha, p), \text{CardMax}(p) = \text{CardMax}(\alpha, p) \]

According to the above definition, we have,
\[ \forall c \in \Psi, \forall d \in \Psi' : \]
\[ c \leq d \Rightarrow (I_{\Psi,d} \subseteq I_{\Psi,c}) \]

Definition 4 (Domain ontology) : Domain ontology is defined by:
\[ O_D = < M_S ; \Psi ; I_{\Psi} ; \Gamma, \Lambda_{\Psi,P} > \]

\( \Psi \) : Hierarchies of concrete classes described by its intension \( I_{\Psi} \)
\( I_{\Psi} \) : Set of the intensions of classes of \( \Psi \)
\( I_{\Psi} = \bigcup_{c \in \Psi} I_{\Psi,c} \)
\( \Gamma \) : Derivation function:
\[ \Gamma = (\Gamma_T, \Gamma_\xi) \]
\( \Lambda_{\Psi,P} = \{ \Lambda_{\Psi,P} \} \) pour \( c \in \Psi \) where \( (\Lambda_{\Psi,P})_c \) is a set of axioms associated to attributes \( P_c \) of \( c \in \Psi \).

The axioms of \( \Lambda_{\Psi,P} \) can be expressed in different manners. One can use rules. For instance, the following rule can be used to define dependency between two attributes \( p \in P_c \) and \( p' \in P_c \) of the same class \( c \in \Psi \) :
\[ \forall o \in \Phi(c), p(o) = v \Rightarrow p'(o) = v' \]
One can also use an equation to define constraints on values of attributes. For instance, the equation below constraints values of the attribute \( \text{Age} \in P_{\text{Young}} \) such that \( (\text{Young} \leq \text{Individu}) \):
\[ \forall o \in \Phi(\text{Young}), \text{Age}(o) \leq 35. \]
4.3 Example
Figure 3, illustrate an example of domain ontology. Here we have:
\[ \text{Node}_\Gamma = \{ \text{Road-Node, Water-Node, etc.} \}, \]
\[ \text{Public} \leq \text{Place} \leq \text{Road-Node}; \]
\[ \text{Manual} \leq \text{Vane} \leq \text{water Node} \]
Note that classes of \( \Psi \) are hierarchically organized.
The top of this hierarchy are classes of \( \alpha_\Gamma = \bigcup \Gamma_\xi (\alpha), \alpha \in \xi \).

5. Inter-ontology relationships
Ontology mapping is important when dealing with multiple ontologies. Many Inter-ontology relationships can be expressed [28, 20].
These relations can be defined at both functional and domain level. Figure 4 illustrates possible examples of relationships between entities.

Relation (R1) : this type of relation correspond to the following different situations:
Two entities of differences theme can cover the same spatial area. It may be the case, for instance, between road entity (an avenue) and an electric network entity (electric wire buried along this avenue) or the case of a building situation (the building itself) and the area on which it is build.
To formalize those relations an algebraic approach can be possible as described in [28]. In order to just point the problem, let’s consider a simple case.
For \( \alpha \in \xi, p \in P_{\alpha} \) such type \( (\alpha, p) = s \in S \), one can define a \( \Sigma \)-homomorphism \( h \) as follow:
\[ h : D_s \rightarrow D_s', \text{ with } \text{Domp}(c, p) = D_s ; \text{Domp}(d, p) = D_s' ; \]
\[ c \in \alpha' \text{ and } d \in \alpha' \]
To illustrate this by an example, consider the abstract class \( \alpha = \text{Person} \), and the attribute \( p \in P_{\alpha}, p = \text{sexe} \).
Given \( \alpha' = \{ \text{Individu, Human} \} \) and suppose that:
\[ \text{Domp} (\text{Individu, sexe}) = D_{\text{sexe}} \]
\[ = \{ \text{Masculine}, \text{Feminine} \} \]
\[ \text{Domp} (\text{Human, sexe}) = D_{\text{sexe}}' \]
\[ = \{ \text{Man}, \text{Woman} \} \]
A possible \( \Sigma \)-homomorphism \( h_{\text{sexe}} \) can be defined as follow:
\[ h_{\text{sexe}} (\text{Masculine}) = \text{Man} \]
\[ h_{\text{sexe}} (\text{Feminine}) = \text{Woman} \]

6. Related word
The work presented in this article concerns the problem of description of ontologies and their relations. It is partially inspired by the work presented in [6,9]. Like [6] we consider the case of an urban application and multi-layered spatial ontologies but in this paper we presents a formalism based on algebraic approach to describe and study the such ontologies. Unlike [9] that described, in general an theoretical manner, an algebraic approach to describe ontologies and their relation, we use the such approach to deal with a concrete example from urban application.

7. Conclusion and future work
In this paper, we present a formalism to describe the multi-leveled ontologies for urban applications. The proposed formalism is based on the fact that the spatial systems can be viewed as comprising several abstract layers, each defining a generic set of functionalities. Each layer contains the entities that share common functional features and are described according to two levels of abstraction. In the functional level, generic ontologies describe entities using abstract classes. In the domain level ontologies are described by concrete
classes. Our ontologies specify entities with attributes to describe their properties and function describing their behavior. Axioms are used to precise the semantics of attributes and functions. We use the ADT to describe the abstract classes. Process derivation allows us to describe the domain specific entities from the generic entities. It consists mainly in interpreting in the domain level the ADT by their models and each abstract class by a set of concrete classes. The latter are the top of the hierarchy of the classes in the domain level. \( \Sigma \)-homomorphism can be used to make relations between entities from the same level. Other types of relations, that are not developed in this paper can be used. Its an interesting issue to explore and to investigate. Another issue to explore is the study of the implementation of proposed formalism using an XML language like OWL. This will enable us to study the limits of this formalism. It is also interest to seek who this framework to describe the multi-leveled ontologies can be use in the other applications, in particularly, the e-learning domain.

References:
[1] Semantic Integration: A Survey Of Ontology-Based Approaches by N.F.Noy. SIGMOD Record, Special Issue on Semantic Integration, 33 (4), December, 2004