ISSN: 2230-9926

Available online at http://www.journalijdr.com

IJDR

International Journal of Development Research
Vol. 10, Issue, 06, pp. 37267-37274, June, 2020
https://doi.org/10.37118/ijdr.19134.06.2020

RESEARCH ARTICLE OPEN ACCESS

A METHOD FOR ONTOLOGY MODELING BASED ON INSTANCES CONCEPTUAL CLASSIFICATION AND FORMALIZATION

Eduardo M. Jorge[1], Uedson S. Reis[3], Antonio C. S. Souza[4], Hernane B. Pereira[1,3], Hugo P. Saba[1,3], Sonia L. Monteiro[2] and Augusto C. Galeo[2]

[1]Department of Exact Sciences, University of the State of Bahia, Cabula, Salvador 41.150-000; [2]National Laboratory of Scientific Computing (LNCC), Quitandinha, Petr´opolis 25.651-075; [3]Postgraduate Programin Computational Modeling and Industrial Technology, SENAICIMATEC, Piatã, Salvador 41.613-036; [4]Federal Institute of Education, Science and Technology of Bahia, Salvador, 40301-015

ARTICLE INFO

Article History:
Received 28th March, 2020
Received in revised form
24th April, 2020
Accepted 06th May, 2020
Published online 30th June, 2020

Key words:
Ontology, M-Mobi, Conceptual modeling, Botton-up modeling

Corresponding author: Eduardo M. Jorge

ABSTRACT

Building a knowledge representation depends on specific skills such as: understanding of modeled reality (it is also called Domain), knowledge of the representation language and, mainly, capacity of abstraction. The process involves such degree of complexity, demanding of the modeling agente an expertise to define axioms (roles) on the conceptual elements, determining constraints and redefining new concepts. This paper presents the proposal of a new conceptual modeling method called Mobi (Instance-based Ontology Modeling) detailing its Process, Editor, Formal Elements and Inference Engine. This method seeks to simplify the representation of abstractions, since the modeling agent builds sceneries through the association of instances and its inference engine identify types of relations and cardinalities.

Citation: ***Eduardo M. Jorge, Sonia L. Monteiro, Augusto C. Galeo, Uedson S. Reis et al.*** "A Method for Ontology Modeling based on Instances Conceptual Classification and Formalization*", International Journal of Development Research*, 10, 06, 37267-37274.

INTRODUCTION

Conceptual modeling is an activity of observing an environment and creating representations in a notation known and accepted by the scientific community. The construction of a conceptual model depends on the understanding of the reality, the model notation of the language, the purpose of the representation (modeling a system, creating a taxonomy, etc.) and the capacity for abstraction by the modeler agent. In the modeling process the modeler agent gradually makes an approximation of reality, encoding it in a notation. According to Kotiadis & Robinson (2008), the agent modeler must determine which aspects should be included, excluded, and the level of detail of what he wants to model and solve a given problem. The research field and proposals for ways of modeling are rich and it depends on the context for

which the model applies. Mathematics and Logic have the basis for the modeling activity, because they present a set of techniques and formal languages appropriate to represent facts of the reality or simulate a natural phenomena. The mathematical formalization ensures accuracy and allows its applicability in a variety of contexts. In the context of Computer Science, an area that uses the mathematical basis, conceptual modeling has also been extensively studied. It has been generating a variety of techniques and languages: (i) the Relational Model proposed by Edgar Frank Codd in 1970 in his famous article "A Relational Model of Data for Large Shared Data Banks" (Codd, 1970); (ii) in the 90s, the UML (Unified Modeling Language) proposed by Booch et al. (1999); (iii) technique to class modeling as Case-based reasoning (CBR) (Watson & Gardingen ,1999); (iv) the Behavior Driven Development (BDD) (Chelimsky ,2010); (v) the test-driven development (Tort et al., 2011). These techniques or languages are not always mutually excludent, they complement each other adding graphical elements of representation to assist in the modeling process. In 1991, the research group DARPA Knowledge Sharing Effort proposed a new method for conceptual modeling called Ontology. It can define as: "An ontology defines the basic terms and relations comprising the vocabulary of a topic area, as well to the rules for combining terms and relations to define extensions to the vocabulary" (Neches & Fikes, 1991). Ontology as a modeling method is based on Logic, and the possibilities of inferences are expressed by a formal language, such as Ontology Web Language (OWL) (Smith et al., 2011). OWL is based on Description Logic which is a subset of Predicate Logic and it has necessary requirements for the modeling process and the building of ontology (Baader et al., 2003). It is well know that the ontology and other modeling techniques are still in evolution process and have incorporated innovations. Regarding the Ontology, its degree of maturity and popularity is lower than other models perhaps due to its recent history. However, there is also a broad set of issues, methodologies and tools that supports Ontology. The most popular methods are: Methontology, Method 101, Cyc, Uschold (Uschold, 1991) (Noy & Mcguinness, 2001).

Beside these, new models are being build by researchers from this area, such as proposed by Bautista-Zambrana (Bautista-Zambrana, 2015), in his paper "Methodologies to build ontologies for terminological purposes". This work has his model focused on the extraction of terms and detection of relations, passing through the build of a terms glossary or a taxonomy, to then build ontologies. Once modeled, an ontology has many applicabilities. In the case of the Semantic Web, Ontology aimed at organizing information through metadata, assigning semantics to the data. There are also applications such as bioinformatics' Gene Ontology Project (www.geneontology.org) that aims to standardize the descriptions of the gene, ensuring that several research groups work on the same vocabulary. The Ontology modeling method has the challenge to reduce the complexity of the modelling process. The person responsible to model the domain area (e.g. bioinformatics) are usually not experts in logic or formal languages. It is important to reduce the complexity of those methods since many formal languages differ from the natural language (Breitman et al., 2006). Reinforcing this issue, Baader et al. (2003) points out the usability of knowledge representation as a critical element in the modeling process of Ontology.

This paper presents a method named as M-Mobi (Method of Modeling Ontology Based on Instances) which is based on the construction of scenarios by describing a set of relations that map the association between a minimum group of instances needed to characterize a relation (reference instances). It also demonstrates the modelling process, the formal specification and the set of elements and rules that composes it. Another issue presented in this paper is the materialization of method into a tool called Web-Mobi Editor. To understand the degree of effectiveness of the method and its editor, this paper also presents brief the implementation analysis of a practical experiment with two groups of participants who modeled Ontology in the Scientific Congress domain. The paper is organized as follows. Section 2 presents the principles, the characteristics, the components, the process and the support tool for the M-Mobi. Section 3 details the formal elements and the inference engine of the M-Mobi. Section 4 is a partially modeled an Election Ontology, illustrating the method and Web-Mobi Editor. The goal is to show how an inference engine assists this process. Section 5 presents the practical experiments applied to M-Mobi. Section 6 presents the conclusions and final measurements on the method M-Mobi.

Principles, Characteristics, Components, Process and Support Tool for M-Mobi: The philosophical principles of the current paradigms of modeling Ontology are based on a strategy "Top-Down" with a focus on generalizations. M-Mobi seeks an approach to not only the "Top-Down" but also "Bottom-Up". It can be observed that the class-based modeling (Top-Down) is common to both, the traditional Object Oriented modeling methods, Relation Entity and the modeling Ontology methods. The consequences of this are that the UML tools such as ArgoUML, Enterprise Architect, Rational Rose, etc. and also the coding ontology editors like Prot´eg´e, Altova SemanticWorks, etc., which are graded in the "Top-Down" paradigm. It is needed to be emphasized that the goal of M-Mobi is not to be a counterpoint to the existing methods and tools, but a "Bottom-Up" modeling method to complement based on class, focusing on the process as a starting point in identifying the relation between groups of instances. The essential feature of the M-Mobi is the possibility of incorporation and/or creation of techniques that minimize the effort of the modeler agent in transforming abstract elements in a formal notation, which means assist her in the process of modeling a domain. In the Fig. 1 is sketched how the M-Mobi acts in the process of modeling Ontology. For this scheme, there is a group of modeling agents divided into two main roles: domain expert and an expert in logic and formal language. These two roles have very different backgrounds and expertise, but they need to interact and act together in this process. The Domain Expert is who holds the knowledge about what should be modeled. For example, a geologist models a structure to represent a set of rocks in a region. In the M-Mobi, the Geologist expert will build and insert association scenarios between instances to reach the general rules of a relation (step (i) in Fig. 1). The model that was built in the M-Mobi is described in a formal language based on Description Logics (step (ii) in Fig. 1).

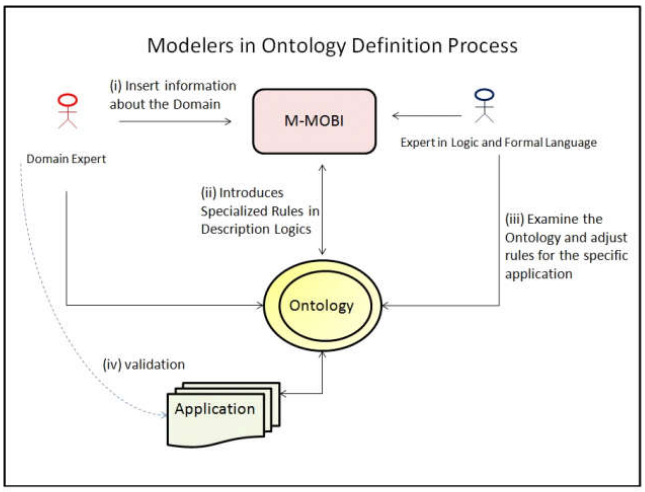

Fig. 1. Interaction between the Moderator Group and M-Mobi

The role of expert in logic and formal language is to analyze the generated Ontology and extend most complex rules, which were not modeled in the previous process with the M-Mobi (step Finally, to validate the model, the domain expert could, for example, analyse whether the generated Ontology meets the needs of application that was generated by checking the assertiveness of the final artifacts of application (step (iv) in Fig. 1). The main component for modeling in M-Mobi's notation is the relation between groups of instances that consists of: (i) two reference groups of instances:

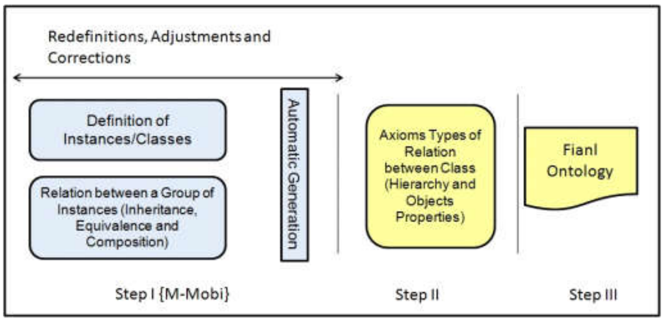

Fig. 2. Process for Ontology Modeling according to M-MOBI

Group A and B, (ii) relations between instances of the groups A and B, called association, (iii) determination of the type of relation that can be Inheritance, Equivalence and Composition. To ensure concision in the process of modeling, the M-Mobi was designed by well-defined formal elements (see Section 3). The formalization allowed transforming abstract ideas and concepts into a model with an explicit notation about rules that ensure the M-MOBI integrity. The M-Mobi is independent of a computational solution and can be used manually. The ontology modeling process of a domain in the M-Mobi follows the scheme of Fig. 2. In the M-Mobi, modeler agent seeks identify and construct the set of relations between instances. An important point in this process is that by applying a set of rules internalized in the M-Mobi (more detail in subsection 3.3), it can automatically get the type of relation (Inheritance, Equivalence and Composition). This aims to help the modeler agent to determine the type of relation and its characteristics, such as domain, range and ownership characteristics (functional, functional inverse, symmetric, etc.) and cardinality. Therefore, by mapping the relations between the instances, it can be deduced the rules and the axioms governing the explicited knowledge in the relation. The tool for to supporting the method is the Mobi Editor which performs the policies of the M-Mobi. The construction of a computational solution for the M-Mobi not only allowed a validation of their specification, but also the application of the method in practical experiments (see Section 5). By using the Editor, the modeler agent can graphically create relations, linking a group of instances through connection in a "Bottom-Up" vision of the model. After assembling the relation, it is shown on the right side of the Editor a class diagram with a "Top-Down" vision of the entire modeled domain. Another feature of the tool is the translation of a model in M-Mobi notation to one in OWL format notation, allowing the interoperability of ontology that was generated from other software and applications. The automation of the method has brought other benefits, such as: reuse of classifications and instances, sharing the modeled scheme between teams, the automatically identification of the relation type by the inference engine and persistence of information.

M-Mobi Formalization

To formalize the M-Mobi various structural components were specified. The method's functional requirements define the fundamental rules that govern all other components, they are:

- RF1 - In the conceptual modeling of a domain, mechanisms must be provided to begin modeling the construction of scenarios, identifying a group of instances that establish general rules (axioms) for the other instances.
- RF2 - In the semantic categorization, an instance must be allowed to be classified in more than one class. In the M-Mobi, for example, an instance of the class "Person" can be linked or unlinked to the classes "Professor", "Student" and "Employee". Techniques to help in naming classes can also be adopted such as (Aguilera et al., 2013).
- RF3 - In the creation of instances, the precept of one unity must be obeyed, whereas in the M-Mobi the instances are unique for all modeled domain and for those that will be modeled.
- RF4 - Through the association between instances, the modeler agent must be assisted to infer the type of relation to be established. The modeler agent will be helped in the choosing process of the relation type. The relation types may vary between inheritance, equivalence or composition.

Based on the requirements and objectives of the method, there were defined the formal elements of the M-Mobi which are based on Description Logics and the set theory. As seen in the previous section, the relations in the M-Mobi are modeling semantic structures composed of a group of instances and its associations. The formal elements of the Mobi for this formalization are grouped by the following categories will be detailed further: domains definition; mapping between domains; the types of relationships that map associations between instances; and inference engine.

Domains definition: Determines sets which group the structural elements of Mobi, instances, classes and domains.

- **Mobi Domain Definition (DDMobi)**: It is the set comprising all instances of the model.

$$DDMobi = \{x_i \text{ such that } x_i \text{ is a instace and } i = 1,...,n\} \quad (1)$$

- **Domain Class (DC)**: It is a set comprising all classes of the model. Each class has a set of instances that satisfy the same characteristics.

$$DC = \{C_j \text{ such that } C_j \text{ is a class and } j = 1,...,n\} \quad (2)$$

- **Domain of Application (DA)**: It is a set that combines specific application domains. Each specific application domain represents a cut of the world or an area of knowledge to be modeled, and it is structured through its Instances, Classes and Relations.

$$DA = \{DAE_l \text{ such that } DAE_l \text{ is a specific application domain and } l = 1,...,n\} \quad (3)$$

Mapping between domains: It is a way of relating Instances with Classes, Classes with Specific Application Domains and Classes with Classes, or a Relation.

- **Mapping Class (C)**: Define the mapping Class (C) as the relation of *DDMobi* in *DC*. This mapping relates instances with their respective Classes. So, a set of instances with the same structural characteristics represents a Class. $C : DDMobi \rightarrow DC$

$$C_j = \{x_k \text{ such that } x_k \text{ are instances that satify the class rules and } k = 1,...,n\} \quad (4)$$

then the instances $x_i \in DDMobi$ should meet the definition of the class rules. Observations on the mapping class (C):

(i) $\forall x_i; i = 1,...,n \rightarrow \exists$ at least one $x_k \in C_j, 1 \leq k \leq n; j = 1,...,n$, that is, a class has to be related to at least one instance.

(ii) The intersection $C_j \cap C_k$ to j 6= k, $j,k = 1,...,n$; can be different of ,that is, the same instance can populate two or more class of DC.

n

(iii) In conclusion the ^{S} C_i is the coverage of *DDMobi*, that is, the instances belonging to all

i=1

classes that represent the set *DDMobi*.

• **Mapping Application (A)**: Define the mapping of Application (A) of *CD* in *AD*. This mapping relates the Classes with their Specific Application Domains. $A : DC \rightarrow DA$

DAE_l = {C_k such that C_k are class that satisfy the rules of Specific Application Domains } (5)

then the classes $C_k \in DC$ then the classes determine the generic structure of an Specific Appplication Domain. Observations on the mapping Application (A):

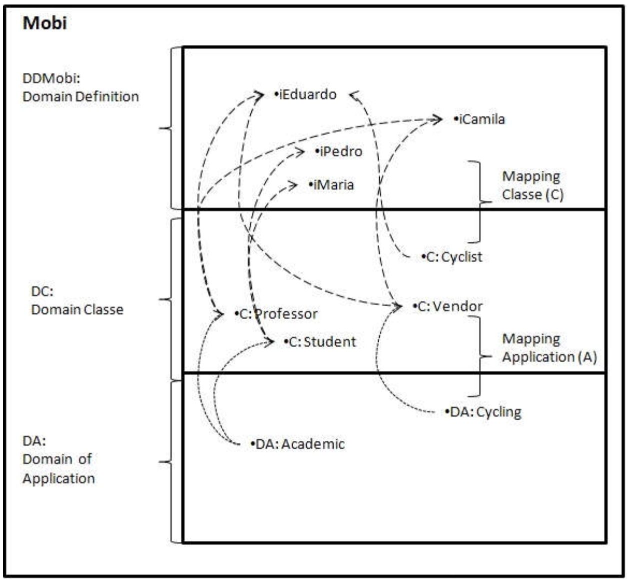

Fig. 3. Example of Mapping Between Domains.

(i) An specific class $C_k \in DC$ populate one and just one Specific Application Domain $DAE_l; l = 1,...,n$.

(ii) On construction of mapping Applicatino (A):$\forall DAE_l; l = 1,..,n; \exists$ at least one $C_k \in DAE_l$.

(iii) How different classes can map the same Instance differents DAE may also contain the same Instance. Either way should be observed that the mapping A defines a partition of the DC domain because:

$$DA \equiv \left\{ \bigcup_{l=1}^{n} DAE_l \wedge DAE_m \cap DAE_n \neq \oslash \right.$$) when m 6= n (6)

- **Mapping Relation (R)**: Define the mapping of Relation (R) of $C_j in C_k$. This mapping relates two Classes within a Specific Application Domains.

$R : C_j \rightarrow C_k$, when that $C_j, C_K \in DAE_l; j,k = 1,...,n$

R = {C_j,C_k such that C_j,C_k are classes that satisfy the rules of relationship of Specific Application Domain} (7)

(i) Can have a self relationship or a endorrelao,then $j = k$.
(ii) A $R : f(x_a) \rightarrow x_b$, being that the instance $x_a \in C_j$ and the Instance $x_b \in C_k; a = b \vee a$ 6= $b; a,b = 1,..,n$.

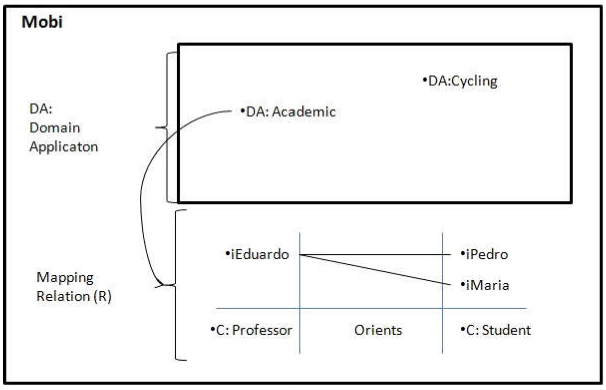

Fig. 4. Example of Mapping Relation to Domain Academic

The Fig. 3 illustrates the use of formal elements of the Mobi and their mappings for specific application domains *Cycling* and *Academic*. Structural elements Mobi for this example are:

- $DDMobi$ = {$iEduardo,iCamila,iPedro,iMaria$}
- DC = {$Professor,Student,V endor,Cyclist$}
- DA = {$Cycling,Academic$}

The mapping elements as the layout of Fig. 3 and the formal notation Mobi are structured in the

$DA = Academic$,then $DAE_Academic$ Mapping Class(C): $C_Professor$ = {$iEduardo,iCamila$} and $C_Student$ = {$iMaria,iPedro$}

$DA = Cycling$, then $DAE_Cycling$ Mapping Class(C): C_Vendor = {$iEduardo,iCamila$} and $C_Cyclist$ = {$iEduardo$}

To illustrate how the formal elements are applied in the construction of a ”Relation”, in Fig. 4 presents an example of mapping a relation to the specific application domain *Academic*.

$DA = Academic$, then $DAE_Academic$ Mapping Relation(C): R : $C_Professor \rightarrow C_Student$ $R : f(x) \rightarrow y$ such that (x,y) = {$(iEduardo,iPedro),(iEduardo,iMaria)$} where $f(x) = Orients$ is a kind of relation of ”Association” of minimum cardinality=1 and maximum=n .

Detailing the types of Relations

There are three types of relation in the M-Mobi: Inheritance, Equivalence and Composition. As it was already mentioned, in modeling of a domain, the modeler agent should define the associations between instances of two classes to determine the general rules of a relation. The hierarchical structures, equivalence, associative and partitive are the form of how the modeler agent can establish relations in M-Mobi. The process is simple and requires the modeler agent to assembly scenarios linking instances in Relations of a domain.

Table 1. Types of function used in the inference engine.

Type	Description	Formalization
Symmetric Group	It is a relation which the origin set is the same as the arrival set. Therefore, the class *Ca* and the class *Cb* are equals.	$R : Ca \rightarrow Cb$, such that $Ca = Cb$
Functional	An instance of a class Ca can only be associated with a maximum of one instance of the class Cb	$R : Ca \rightarrow Cb$, such that $(\forall Ia \in Ca)(\forall Ib_1 \in Cb)(\forall Ib_2 \in Cb)(r_1(Ia,Ib_1) \wedge r_1(Ia,Ib_2) \rightarrow Ib_1 = Ib2)$
Injective	An instance of the class Cb can only be associated with a maximum of one instance of the class Ca	$R : Ca \rightarrow Cb$, such that $(\forall Ia \in Cb)(\forall Ia_1 \in Cb)(\forall Ia_2 \in Cb)(r_1(Ia_1,Ib) \wedge r_1(Ia_2,Ib) \rightarrow Ia_1 = Ia_2)$
Total	No instances can remain unassociated in the class Ca, i.e., all instances of the class Ca belong to a relation	$R : Ca \rightarrow Cb$, such that $(\forall Ia\ Cb) \in Ca)(Ib \in (r_1(Ia,Ib))$
Surjective	No instances can remain unassociated in the class Cb, i.e., all instances of the class Cb belong to a relation	$R : Ca \rightarrow Cb$, such that $(\forall Ib \in Cb)(Ia \in Ca)(r_1(Ia,Ib))$
Equal Instances	There is only association between the same instances	$R : Ca \rightarrow Cb$, such that $(\forall Ia \in Ca)(\forall Ib \in Cb)(\forall r_1(Ia,Ib) \rightarrow Ia = Ib)$

Table 2. Result of students master about the process of Modeling Ontology on "Process of Submission and Evaluation of Articles of Scientific Events".

Participants	Modeling in Prot´eg´e	Web-MOBI Editor
Person 1	low (10/36)	average (18/36)
Person 2	average (13/36)	average (17/36)
Person 3	low (6/36)	low (10/36)
Person 4	average (13/36)	high (27/36)
Person 5	low (12/36)	average (16/36)
Person 6	low (8/36)	average (14/36)
Person 7	low (10/36)	average (13/36)
Person 8	low (12/36)	average (16/36)
Person 9	low (9/36)	low (13/36)
Person 10	low (8/36)	low (10/36)
Person 11	low (12/36)	average (16/36)

Table 3. Result of students PhD about the process of Modeling Ontology on "Process of Submission and Evaluation of Articles of Scientific Events".

Participants	Modeling in Prot´eg´e	Web-MOBI Editor
Person 1, 2 and 3	average (16/36)	average (15/36)
Person 4 and 5	high (26/36)	high (26/36)
Person 6 and 7	average (15/36)	average (20/36)
Person 8 and 9	average (17/36)	high (11/36)
Person 6 and 7	average (15/36)	average (20/36)

Therefore, each type of M-Mobi's relation are detailed. *R hierarchical* is a type of Relation that given two Classes *Ca*,*Cb*, it is possible to assert that exists the Relation *Ca pInheritance Cb*, i.e., *Cb kind−of Ca* if, and only if, *Ca* 6= *Cb* and there is at least one relation that links instances which are equal and belong to the classes *Ca* e *Cb*. For *Ca* class to be defined as a class of a highest hierarchy (a superclass), there must be some element (instance) of *Ca* unrelated to *Cb*. This last question is what differs a *R inheritance* Relation of *R equivalence* Relation, but the relation with the condition of being a *R hierarchica* will never be a *R composition*. *R equivalence* is a relation to model Classes which are synonyms. Given two classes *Ca* and *Cb*, it is possible assert that exists *Ca pEquivalence Cb*, i.e., *Ca equivalent Cb*, if and only if *Ca* 6=Cb and there is at least one Relation that links Instances which are equal and belong to the classes *Ca* and *Cb*. Furthermore, the scenario that was built must configure that all instances of the class *Ca* should be associated with the class *Cb* and all instances of the class *Cb* should be associated with the class *Ca*. In this case the relation cannot be determined automatically without the help of the modeler agent, since the relation can be an equivalence or (exclusive) inheritance. So, it is borne by the modeler agent to choose the type of relation.

It is observed that the relation with the condition of being a *R equivalence* will never be a *R composition*. *R composition* is a way to model the relationship between two classes through partitive and associative relation. In this type of relation the modeler agent indicates a "name" to describe the semantics of the relation, unlike the inheritance and equivalence relations which the name is set by default as is-a and equivalent. Another particular element of this type of relation is the cardinality, which is obtained through the inference engine (more details section 3.4) based on the rules of the M-Mobi. The cardinality is defined as the minimum and maximum number of elements in a Relation. For example, for a composition Relation *Ca has Cb*$(1,n)$, the modeler agent identified it with the name="has". The cardinality (minimum, maximum) was obtained through a set of relation between instances of the classes *Ca* and *Cb* established by the modeler agent. *R composition* is typified in four ways: unidirectional functional, unidirectional, bidirectional and symmetrical. Hereafter, each one of them is detailed.

- *R composition*(*unidirectional functional*) is a relation in which given two classes *Ca* and *Cb*, it can be affirmed that there is a composition relation *R composition* (*unidirectional functional*) if and only if whatever the Instance of the class *Ca* it is related to one and only one Instance of class

- *R composition* (*unidirectional functional*)) is a relation in which given two classes *Ca* and *Cb*, it can be affirmed that *R composition* (*unidirectional*) if and only if exists a Instance of the class *Ca* that is related to several Instance of the Class *Cb*.
- *R composition*(*bidirectional*) is a relation that has an inverse, i.e., given two classes *Ca* and *Cb*, it can be affirmed that it is a bidirectional composition if and only if there is an outward relation between Instances of the classes *Ca* and *Cb* and there is a backward relation between Instances of classes *Cb* and *Ca*. In this type of relation the modeler agent should indicate the name of the outward Relation and the name of the backward Relation, i.e., a name to identify the Relation from *Ca* to *Cb* and another name to identify the Relation from *Cb* to *Ca*. For example, name=”has” and name inverse =”belongs to”, i.e., *Ca has Cb* and its inverse *Cb belongs to Cb*. A relation *R composition* (*bidirectional*) decomposes into two unidirectional properties, so all the settings made for composition, unidirectional functional and unidirectional relation are valid for bidirectional.
- *R composition* (*symmetrical*) is a Relation that respects the same characteristics as the bidirectional composition relation with the particularity of identifying the outward and the backward names, therefore name = name inverse.

In the next section, we explain how the M-Mobi is structured to infer the types of relations based on scenarios provided by the modeler agent.

The inference engine of the M-MOBI: The main goal of the inference engine of the M-Mobi is to identify if a relation is of type *R hierarchical*,*R equivalence* and *R composition*. For this, the inference engine consists of a set rules created based on the types of functions: symmetric group, Functional, Injective, Total, surjective and Equal Instances (details in Table 1). This section is limited to present four rules of inference engine. The M-Mobi is based on the concept of mathematical induction which in principle searches through a partial specification to determine a general rule. In the case of the M-Mobi, the idea is to infer the types of relations of an ontology based on the modeled scenario and from associations between instances proposed by the modeler agent. Fig. 5(a) and Fig. 5(b) are presented four rules of the inference engine described by combining several types of functions presented in Table 1. The rules show how two sets of scenarios of association between instances can be used to achieve the type of relation. The rules R1 and R2 act in the Inheritance and Equivalence Relations respectively. For these two rules, there is a condition [Equal Instances] (There is only association between the same instances) combined with a ”no [Symmetric Group]” (It is not a relation which the origin set is the same as the arrival set). The combination of these two types of functions determine that a scenario with these model is not characterized as a composition, i.e., or it is an equivalence or it is inheritance. In the rule R2, the way to differentiate the Inheritance Relation from Equivalence Relation is the occurrence of the type of function ”no [total]” or (exclusive) ”no [Surjective]”. Therefore, it denotes a condition where there is at least an unassociated Instance in the Class *Ca* or (exclusive) there is at least an unassociated Instance in Class *Cb* (rule R2). Therefore, when there is a condition and the others are kept, the inference engine determines that the Relation is an Inheritance. Fig. 5(c) and Fig. 5(d) show two scenarios of association between instances that determine by using inference the composition relation and its cardinality (minimum and maximum). In the inference engine, the type of function ”no [Equal Instances]” (there was no association between instances equal) is used to determine that this relation is a composition (rule R3). Since the type of function [total] and [Surjective] determine the minimum cardinality of a Relation.

(a) Rule R1 - Inheritance and (b) Rule R2 - Inheritance. Equivalence.
(c) Rule R3 - Composition. (d) Rule R4-Composition.

Fig. 5. Rules for inferring the Inheritance, Equivalence and Composition Relation.

Detailing the rules to infer the types of relationship: Inheritance, Equivalence and Composition:

- Rule R1 Inheritance and Equivalence (Fig. 5(a)) if (There is only association between the same instances [Equal Instances]) and (No instances can remain unassociated in the class *Ca* [Total]) and (No instances can remain unassociated in the class *Cb* [Surjective]) and no (It is a relation which the origin set is the same as the arrival set [Symmetric Group]) Then *Ca kind of Cb*, *Cb kind−of Ca* ou *Ca equivalent Cb*
- Rule R2 Inheritance (Fig. 5(b)) if (There is only association between the same instances [Equal Instances]) and no (No instances can remain unassociated in the class *Ca* [Total]) and (No instances can remain unassociated in the class *Cb* [Surjective]) and) and no (It is a relation which the origin set is the same as the arrival set [Symmetric Group]) Then *Cb kind − of Ca*
- Rule R3 Composition ((Fig. 5(c)) if no (There is only association between the same instances [Equal Instances]) and not (No instances can remain unassociated in the class *Ca* [Total] and (No instances can remain unassociated in the class *Cb* [Surjective]) and (An instance of a class *Ca* can only be associated with a maximum of one instance of the class *Cb* [Functional]) and (An instance of the class *Cb* can only be associated with a maximum of one instance of the class *Ca*[Injective]) Then $Ca\ \{name\}\ Cb(0,1)$, $Cb\ \{name_{i}nverse\}\ Ca(1,1)$
- Rule R4 Composition (Fig. 5(d)) if no (There is only association between the same instances [Equal Instances]) and (No instances can remain unassociated in the class *Ca* [Total] and (No instances can remain unassociated in the class *Cb* [Surjective]) and no (An instance of a class *Ca* can only be associated with a maximum of one instance of the class *Cb* [Functional]) and (An instance of the class *Cb* can only be associated with a maximum of one instance of the class *Ca*[Injective]) Then

$Ca\ \{name\}\ Cb(1,n)$, $Cb\ \{name_{i}nverse\}\ Ca(1,1)$

For example, in rule R3 there is a conjunction of two types of function ”no [total]” and [Surjective], i.e., there are unassociated Instances in class *Ca* and and all instances in Class *Cb* are associated, which provides that the minimum cardinality is between 0 *Ca* and *Cb* and, in the backward of the relation, the minimum cardinality is 1 (one), thus *Ca* [*name*] *Cb*(0,?) and *Cb* [*inverse name*] *Ca*(1,?). Therefore, when there are elements not associated with the minimum cardinality, it is set as the types of function [Functional] and [Injective] determine the maximum cardinality of a relation. For example, in rule R3 all instances of the class *Ca* has the maximum of 1 (one) association with Instances of Class *Cb* and all Instances of Class *Cb* has the maximum of 1 (one) association with Instances of class *Ca*, i.e, it is a [Functional] and [Injective] relation. The maximum cardinality in this scenario has the value 1 (one), so the relations of the rule R3 are *Ca* [*name*] |*Cb*(0,1) and *Cb* [*inverse name*] *Ca*(1,1). Similar to the rule R3, the rule R4 is a composition having a specificity of having a function type "no [Functional]" which determines the maximum cardinality towards *Ca* to *Cb* has value n (many), so for R4 the Relation is *Ca* [*name*] *Cb*(1,*n*), *Cb* [*inverse name*] |*Ca*(1,1).

Example of Modeling by using the Inference Engine: This section is partially modeled the Election Application Domain through the tool Web-Mobi Editor, demonstrating how the inference engine supports the modeler agent in this process. The Relations modeled in the tool exemplify the rules R1, R2 and R4 which were detailed in the previous section. The rule R3 is not going to be shown, because it is similar to composition on R4, varying only what concerns about cardinality. For the Election domain is going to be modeled three relations:

- Party equivalent PoliticalGroup, exemplifying the Rule R1(Equivalence);
- Mayor kind-off Candidate, exemplifying the Rule R2 (Inheritance);
- Party has Candidate (0,n) and inverse Candidate affiliate Party (1,1), exemplifying the Rule R4 (Composition);

The modeler agent initiates the modeling, in this example, by associating instances of parties and classifying as Party and PoliticalGroup. Automatically, the first association r1(PartyC, PartyC), the Editor already eliminates the option of composition of the relation types, due to existing a combination of [Equal Instances] and "not [Symmetric Group]". In this case, the modeler agent is who decides whether it is an equivalence or inheritance. The types of function and the elements of the relation Party equivale PoliticalGroup are: If {[Equal Instances] r1(iParty 1,iParty 1), r2(iParty 2,iParty 2) e r3(iParty 3,iParty 3)} and {[total] - No instances can remain unassociated in the class Party} and {[Surjective] - No instances can remain unassociated in the class PoliticalGroup} and {"no [Symmetric Group]" - The class Party is different class PoliticalGroup } Then Party equivalent PoliticalGroup. (Observe that in this case the agent modeler had to choose between Inheritance and Equivalence.) After that the modeler agent sets up a association scenario between instances applying rule R2 (Inheritance) presented Fig. 5(b) . In this case, the M-Mobi automatically indicates that the relation is inheritance. The types of functions and elements of the relation Mayor kind off Candidate are: If {[Equal Instances] r1(iMayor 1,iMayor 1) e r2(iMyor 2,iMayor 2)} and {"no [Total]"- Instances iViceMayor 1 and iViceMayor 2 class Candidate are not associated} and {[Surjective] - No instances can remain unassociated in the class Mayor} and {"no [Symmetric Group]" - The class Candidate is different class Mayor} Then Mayor kind-off Candidate. Finally, the modeler agent sets up a Composition Relation scenario between instances adhered to rule R4 (Composition), generating an outward *Party has Candidat* (1,*n*) and backward *Candidate affiliate Party* (1,1).

Validation of the M-Mobi in a Practical Experiment: A controlled experiment was performed in order to validate the M-Mobi and it was structured through three steps in the following order:

- A presentation about concepts of Conceptual Ontology Modeling to our participants. All concepts was presented without any tools.
- A presentation about concepts of Conceptual Ontology Modeling to our participants. This time it was presented with the Prot´eg´e tool (method 101) and Web-MOBI Editor.
- The participants modeled an Ontology for the process of submission and evaluation of articles of scientific events using the Prot´eg´e tool and Web-M-MOBI Editor tool.

In the process of data collection was used triangulation strategy to obtain information through research questionnaires, review of the modeled elements and observation of the researchers. The focus groups were: (i) students master SENAI CIMATEC discipline Computational Modeling Program and Industrial Technology on 17, 18 and 19 June 2010; (ii) students of the Multi-institutional and multidisciplinary PhD in Knowledge Dissemination (UFBA, LNCC, UNEB, UEFS UFABC, IFBA and SENAI-CIMATEC) in the discipline "Representative System of Knowledge" on 17 and 24 September and 01 October (2010) For the realization of practical experiment the modeling area was the Ontology for the Submission Process and Scientific Events Articles Assessment. The result for group of 11 students of the master's program were: (i) in Method 1010 / Prot´eg´e, 9 reached the low assertiveness level and 2 reached the average level; (ii) in the M-MOBI, 2 participants reached the low assertiveness level 8 reached the average level 1 and reached the highest level. escrever que no questionrio de entrevista dodos se consideravam intermedirios em relao a atividade de modelagem conceitual. Processo foi realizado para alguns alunos em duplas para o mestrado O gabarito indicoi 36 elementos de modelagem entre identificao de classes e relaes Escrever sobre as faixas de assertividade [1, 12] baixo; [13, 24] mdio; [25, 36] alto In the tablet 2 present the result, summarize as: For the 1010 / Prot´eg´e method ,nine (9) students reached the low assertiveness level and two (2) reached the average level. For the M- MOBI method ,two (2) participants reached the low level of assertiveness ,eight (8) reached the average level and one (1) reached the highest level. In the tablet 3 present the result, summarize as: (i) A for method 1010 / Prot´eg´e ,no student has reached the low level of assertiveness ,seven (7) students reached the average level and two (2) students reached the highest level; (ii) For the M-MOBI ,two (2) students reached the low level of assertiveness ,five (5) students reached the average level and two (2) students reached the highest level. The conclusion about the assertiveness' level, making a qualitative analysis through the observations of the researchers, is that the two methods have similar results. In the quantitative analysis the MOBI's result in the experiment proved better to the masters students in 55% of the modelings, the method 101 was better in 9% of the modelings, and in 36% of the modelings both had equality, see details in Table 2. In the quantitative analysis the MOBI's result in the experiment proved better to the doctoral students in 11% of the modelings, and in 89% of the modelings both the methods were equals, see the results in details in Table 3. Participants reported that the M-MOBI helped in the validation of the relations due to construction of the scenarios. Many participants related that they used the strategy of scenery's building in your mind during the modeling's process. Many participants related that they used the mental strategy of building the scenery during the modeling's process. Therefore, they concluded that the M-MOBI materializes one step performed mentally, making easy the design of the model and reducing the abstraction's level.

Final Considerations

In this article, we have presented a novel method for ontology modeling based on instances, detailing its process, formal elements and inference engine. To illustrate the applicability of the method and its feasibility was described an example of modeling, by a computational solution called Web-Mobi Editor, though which an essential feature of the proposed model, which differs from the methods and tools focused on the class paradigm (top-down), is the use of a component (Relation) for association between a group of instances as a basic element in the modeling process, i.e., a Bottom-up strategy. Therefore, it was sought to develop a simple method where the modeler agent (expert in the domain) creates the model by connecting only instances, materializing abstractions in the scenarios. To guide the modeler agent, it was produced an inference engine with the method's premises. In the Web-Mobi Editor, a type of relation is identified automatically (if it is Inheritance, Equivalence or Composition).

In the case of a composition relation, it is even determined its cardinality. Thus, the tool utilizes the Relations automatic generation to build a vision (Top-down) of the model by merging the Bottom-up and Top-down strategies. Finally, it is emphasized the area of research in the modeling field, a spot that is still opened is the strategy of how it is possible to evolve the ontology, attending to the natural process dynamics of the understanding enrichment of a domain. Changes in the modeling process are common and it may occur due to variations in the domain or new mediation made by the modeler agent. This is another innovative point of the method that enables this dynamic of the knowledge evolution of a modeled domain enunciated by Liang (2005). In the M-Mobi, the instances are unique and the occasional mutability is made only in the context of their classification. The M-Mobi allows through a change of scenery (associations between instances), the classes' structures and their axioms can change without much effort for modeler agent. This is due to the process of automatic generation of the structures of a model through the M-Mobi's inference engine.

Finally, it is expected that the impact of the proposed of the M-Mobi can influence the emergence of new methodologies and editors and or extension of the existing ones. Future research should to test the M-Mobi's conceptual schema, the work of Tort and Oliv Tort & Oliv´e (2010) provides some approaches. We also propose a future work for extension of the M-Mobi aiming a development environment that integrate modeling and programming, permitting to user develop a generic and platform independent business rule. So, it is necessary creating a programming language to the M-Mobi and use a process converting the model and the logic into specific languages, the works (Luo et al., 2013) and (Ravent´os & Ovil´e, 2008) show some ways to do it.

REFERENCES

Aguilera, D., G´omez, C., Oliv´e, A. 2013. A complete set of guidelines for naming UML conceptual schema elements. Data & Knowledge Engineering, Vol. 88, p. 60-74. Elsevier B.V.

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D. & Patel-Schneider, P. F. 2003. The Description Logic Handbook: Theory, Implementation and Applications, Cambridge University Press.

Bautista-Zambrana, Maria, 2015. Methodologies to Build Ontologies for Terminological Purposes. Procedia - Social and Behavioral Sciences, volume 173, pages 264-269.

Booch, G., Rumbaugh, J. & Jacobson, I., (1998) The Unified Software Development Process, AddisonWesley, Redwood city.

Breitman, K., Casanova, M. A., Truszkowski, W. 2006. Semantic Web: Concepts, Technologies and Applications. NASA Monographs in Systems and Software Engineering. Springer-Verlag New York,Secaucus, NJ, USA.

Chelimsky, D., Astels, D., Dennis, Z., Hellesy, A., Helmkamp, B., North, D., (2011) The RSpec Book: Behaviour-Driven Development with RSpec, Cucumber, and Friends. The Pragmatic Programmers, LLC.

Codd, E. F., 1970. A Relational Model of Data Large Share Data Banks. Communicatios of the ACM 13, 377-387.

Jorge, E. M. F. 2010. M´etodo de Modelagem de Ontologia Baseado em Instncias. Texto Para Qualificao (Doutorado). UFBA/INCC/UNEB/UEFS/UFABC/IFET/SENAI-CIMATEC, Salvador, Brasil.

Kotiadis, K. & Robinson, S. 2008. Conceptual Modelling: Knowleged Acquisition and Model Abstraction. Proceedings of the 2008 Winter Simulation Conference, 952-958.

Liang, Y., F. 2005. Ontology Versioning and Evolution for Semantic Web-Based Applications. Tese (Intelligence, Agents and Multimedia Group School of Electronics and Computer Science Faculty of Engineering and Applied Science) — University of Southampton, Southampton.

Luo, R., Peng, X., Lv, Q., Wu, M., Peng, B., Wang, S., Guo, M. 2013. An MDA Based Modeling and Implementation for Web App. Journal of Software, Vol. 8 (8), Academy Publisher.

Noy, N., F., & Mcguinness, D., L. 2001. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Tech. Rep. KSL-01-05 and Stanford Medical Informatics Tech. Rep. SMI-2001-0880.

Neches, R., Fikes, R. E., Finin, T., Gruber, T., Patil, R., Senator, T. & Swartout, W. R. 1991. Enabling Technology for Knowledge Sharing. AI Magazine 12, 36-56.

Ravent´os, R., Oliv´e, A., (2008) An object-oriented operation-based approach to translation between MOF metashemas. Data & Knowledge Engineering 67 (2008) 444-462. Elsevier B. V.

Smith,M.K.,Welty,C.,Mcguinness,D.L.,2011OWLWebOntologyLanguage. http://www.w3.org/TR/owl-guide/.

Tort, A., Oliv´e, A., Sancho, M., 2011 An approach to test-driven development of conceptual schemas. Data & Knowledge Engineering Vol. 70, p. 10881111. Elsevier B.V.

Tort, A., Oliv´e, A., 2010. An approach to testing conceptual schemas. Data & Knowledge Engineering Vol. 69 (2010) 598618, pp. 1-3. Elsevier B.V.

Uschold, M., King, M., 1991. Towards a Methodology for Building Ontologies. Internationl Joint Conference on Artificial Intelligence. Montral, Qubec, Canada, n. 12, p. 36-56, 1991.

Watson, I., Gardingen, D., 1999. A Web Based Case-Based Reasoning System for HVAC Sales Support. Applications & Innovations in Expert Systems VI. R. Milne and A. Macintosh and M. Bramer, Springer, London, 11-23.

****