CONCEPT CATEGORIZATION IN PRE-CONCEPTUAL SCHEMAS

Carlos Mario Zapata J.[1], Oscar Pastor[2], Giancarlo Guizzardi[3], Renata Guizzardi[3]

Universidad Nacional de Colombia, cmzapata@unal.edu.co
Universidad Politécnica de Valencia, opastor@dsic.upv.es
Federal University of Espírito Santo, {gguizzardi, rguizzardi}@inf.ufes.br

Abstract. Pre-conceptual schemas are used in software engineering for automatically generating conceptual diagrams and source code of a given domain. Inside pre-conceptual schemas, concepts are not currently categorized, leading to misuses of such nodes. From the definition of Aristotle's Categories, many authors have discussed and recognized the need of classifying elements into categories. Ontologies take advantage of such feature and establish complete taxonomies of categories for using in several domains. In this paper, we exemplify the use of an ontological theory for extending the meaning of concepts inside pre-conceptual schemas, in order to recognize the differences among several types of concepts for representing the knowledge related to a certain domain by means of pre-conceptual schemas. The proposed concept categorization is illustrated by a running example.

Keywords: Pre-conceptual schemas, knowledge representation, concept categorization, ontological classification

1 Introduction

Pre-conceptual schemas [1] have been used to represent concepts and relationships belonging to a certain domain, searching for the automated generation of UML conceptual schemas and source code of a software application [2]. Pre-conceptual schemas employ linguistic definitions for every node, trying to get stakeholder understanding and validation by using a controlled language, which is closer to the stakeholder discourse. However, concepts (a special kind of nodes) inside pre-conceptual schemas have no distinction in their essence and they can be used with one-single constraint: concepts are nouns or noun phrases of the stakeholder discourse. The strong differences we can argue between concepts are not used inside the semantics of pre-conceptual schemas, leading to either semantic or pragmatic misuse of such elements.

Concepts can be treated differently in the same way that Aristotle made a clear distinction of objects of human apprehension and classified those under ten categories, namely: substance, quantity, quality, relation, place, time, position, state, action, and affection [3]. Since the appearance of this work, categorization has been the focus of many essays and research projects [4, 5, 6], searching for the common ground in which categories of elements can be understood. Related to knowledge representation, ontologies have been used to categorize the concepts inside certain domains [7]. Consequently, domain ontologies emerge to classify terms (concepts) and relationships belonging to a given domain by means of ontological categories.

Conceptual modeling can take advantage from ontologies in order to establish several categories of the concepts used in this context. Hence, Guizzardi [8] proposes an extension of UML (Unified Modeling Language) by using ontological theories. The UML meta-model is, then, redesigned by taking into account a number of ontological distinctions, as a way to improve the semantics behind conceptual classes. A research assumption of this paper is that the benefits earned by the use of ontology techniques in conceptual modeling can also be carried to pre-conceptual schemas.

As a way to deal with the misuse of concepts inside pre-conceptual schemas, in this paper we illustrate an extension of the syntax and semantics of such schemas, by incorporating some distinctions between the concepts we can use when representing a certain domain. In order to complete this task, we employ the work of Guizzardi [8] in terms of ontological concept categorization in pursuing a new way to represent and use concepts in terms of their categories inside pre-conceptual schemas. Our aim is to semantically improve the pre-conceptual schemas, making them clearer and more precise. The claim is that enriched pre-conceptual schemas can be more efficient and effective in terms of the concept categorization.

The remainder of this paper is organized as follows: section 2 is devoted to the theoretical framework in which this proposal is based; concept categorization in several contexts is discussed in section 3; in section 4 we propose a new way to establish differences between concepts in pre-conceptual schemas and we exemplify the proposed categorization with a running example; conclusions and future work are presented in section 5.

2 Theoretical framework

Pre-conceptual schemas were created to solve some problems in early validation of the requirements of a software application [1]. When interviews are conducted for gathering the information from a problem domain, stakeholders and analysts can create a pre-conceptual schema for representing such information. Then, such schema can be translated into several diagrams for software development, *e. g.* UML diagrams [1], KAOS goal diagram [9], and OO-Method diagrams [10]. Such kind of transformation work from either natural or controlled language can be traced back to several projects like NIBA [11], NL-OOPS [12], and CM-BUILDER [13]. Also, the work of Moreno *et al.* [14] can be a source of information about automated generation of conceptual schemas from natural language. However, from the main perspective discussed in this paper, all these projects treat concepts in the same way, making no difference between them, except by the typical class-attribute dichotomy stated by many authors.

Figure 1 shows an example of pre-conceptual schema, in which rectangles are concepts (reserved for nouns and noun phrases), ovals with solid lines are structural relationships (“to be” and “to have” verbs), and ovals with dotted lines are dynamic relationships (reserved for dynamic verbs that resemble activities). Thin arrows are connections between concepts and dynamic/structural relationships, and *vice versa.* Thick arrows are implications and represent cause-and-effect associations between dynamic relationships. Note that the pre-conceptual schema in Figure 1 can be simply read by following the connections and creating 3-element phrases, such as: “pet has identification” or “owner requests appointment.” In the case of implications, phrases must be built by reading two 3-element phrases, such as: “if veterinarian auscultates pet, then veterinarian registers diagnosis”.

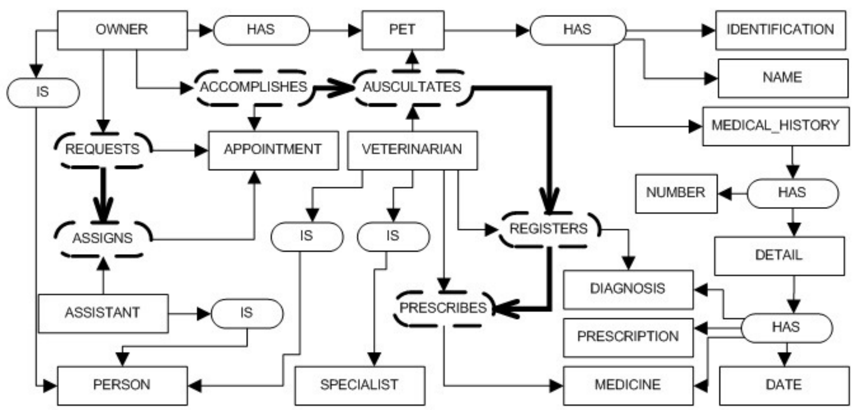

Fig. 1. An example of pre-conceptual schemas.

3 Concept categorization

The Aristotle’s ten categories are considered the beginning of such kind of concept classification. The discussion about subject, essence, and accident is the foundation for the categories stated by Aristotle, as an attempt to understand the matter of things and the way we can classify concepts [3]. Aristotle stated that concepts are essentially different and their properties are dependent on the categories in which they are classified [3].

Roth [4], in terms of “classifying” as a human activity to understand the world, advocates that such activity tends to minimize contradictions among several pieces of information. In Roth’s words [4], scientists tend to classify because concepts have “perceptual distinctions whose reconstruction is facilitated by their membership in a class with certain properties.” (pp. 583). Following this line of thought, categories are important to communicate and study the features of some elements, as a way to decompose their complexity. Roth [4], also, discusses the use of classification in computer-driven environments, such as databases and repositories, reinforcing the idea of communicating concepts between machines and humans.

Hierarchies are commonly the focus of categorization [5]. We can use hierarchies to create sorted sets of elements which, in turn, can be useful for inferring the position of similar elements in the hierarchy. In this case, cognitive rules play a crucial role for determining the whole composition of the hierarchy. Yang *et al.* [5] establish that "Human cognition relies on such a hierarchical representation of objects in the process of perceiving and understanding, and can perform object categorization at different levels of the hierarchy." Also hierarchical in nature, *LexikoNet* [6], a large lexical hierarchy of German nouns, exploits the usage of roles and types, in addition to meronymic and holonymic relations. *LexikoNet* is intended to be used in lexicographical analysis, since it is based on conceptual classes to provide the needed abstraction for this task. In the same line of thought, Körner and Brumm [15] exploit several ontologies to clarify the senses of natural language specifications.

When hierarchies are requested and used, ontologies are the natural answer to these requests [7]. In the context of concept categorization, a large amount of concepts can be represented and categorized by means of domain ontologies. Stuckensmidth and Klein [7] propose a structure-based partition of large concept ontologies in order to solve some common problems of this kind of ontologies. For example, the maintenance of a non-partitioned large ontology is very difficult, since domain experts can be world-wide spread and the lack of structure create barriers to the ontology maintenance process. Other problems are related to the validation, use, and processing of such ontologies, if partitioning is not accomplished on them, because the size and the lack of structure prevent the user to employ the large ontology as a set of smaller ontologies. Concept categorization can support the solutions to these problems by labeling the large ontology and then cutting it into smaller ones.

Conceptual modeling is one of the areas that can be supported by ontologies. In fact, Guizzardi [8] presents a complete ontology-driven analysis of the structural part of UML conceptual models. In this context, classifiers (taken here as synonyms of concepts in terms of the domain discourse) are extended in order to fit a detailed semantics, by adding new inherited sets of stereotyped classes, intended to act as categories. A small fragment of the proposed extension can be seen in figure 2. The grey meta-classes are concrete and represent ontological categories of object types represented as newly added distinctions among UML classes.

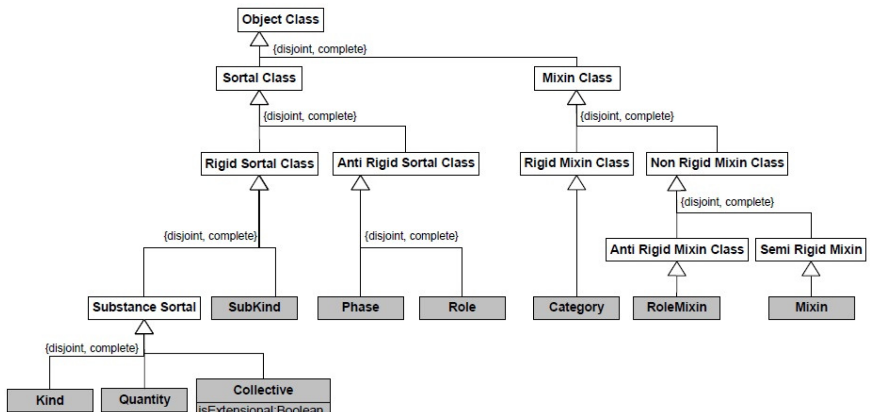

Fig. 2. Part of the proposed extension to UML diagrams for ontological purposes [8].

Pre-conceptual schemas are knowledge representations of the domain discourse, in which we can recognize some of the most common constructs belonging to UML diagrams [1]. Like UML conceptual models, pre-conceptual schemas have concepts and relationships, revealing a common foundation between these two knowledge representation mechanisms. Inside a pre-conceptual schema, concepts are equally treated, because the only constraint to concepts is related to the linguistic condition of being either nouns or noun phrases. However, constraints can be defined in terms of the usage of concepts. For example, in figure 1 there is no constraint to prevent that phrases like “clinic auscultates pet” can be defined. In fact, the stakeholder can manifest such phrase, but empirically, we can discover that the actions must be assigned to people, in order to recognize responsibilities in performing actions. Similarly, we can discuss the role of terminal concepts as repositories of data, instead of non-terminal concepts or root concepts, which cannot receive values. For example, the concept “name” is supposed to save something like “Lassie” or “Flipper” and we know it, but it is not so clear what the value of “pet” itself is. Maybe we can use “dog” or “friend” as values for the concept “pet” and, in this case, we could misuse the concept. Currently, pre-conceptual schemas are affected by the lack of differentiation among concepts, leading to problems in terms of their semantic and pragmatic interpretation. The assumption here is this fact makes pre-conceptual schemas good candidates to be enriched by means of a foundational ontology, as proposed by Guizzardi [8].

4 Ontological categorization of pre-conceptual schema concepts

4.1 Conceptual schemas and ontologies

Pre-conceptual schema concepts can be translated into several constructs of UML. Depending on the rules we can apply, we can specify that concepts are either classes or attributes from class diagram, objects from communication diagram, and objects to be analyzed by means of a state machine [1]. Since classes/attributes are similar modeling primitives to the ones present in pre-conceptual schemas, we take that the UML extension proposed by Guizzardi [8] and depicted in figure 2 can, in principle, be applied to pre-conceptual schema concepts. The aim of such work is to inform real-world semantics, and thus ontological foundations to the pre-conceptual schemas. If we can characterize the concrete meta-classes of figure 2 to define the main features of pre-conceptual schema concepts, we can clearly differentiate among these concepts and we can assign them the right meaning, trying to avoid misuses of the phrases we need to incorporate in the pre-conceptual schema. This ontological foundation of pre-conceptual schemas is the strategy we follow in section 4.2.

4.2 Definition of concrete meta-classes from the UML extension in terms of pre-conceptual schemas

Guizzardi [8] describes, by means of profiles, the main features of the ontological meta-classes added to the meta-model of UML. We follow here a similar line of thought for representing the main features of the ontological concept categorization of pre-conceptual schemas.

Previously, we must define the way we can graphically represent the type of concepts. Since stereotypes constitute the UML standard way to express extension and even though pre-conceptual schemas are not UML-type diagrams, for the sake of clarity we propose to use the same symbols ("〈〈" and "〉〉") to represent extension in the features of concepts inside a pre-conceptual schema. These symbols will be used inside the rectangle that represents the concept and prior to the name of the concept. Between the symbols, we will locate the name of the label that can be assigned to the concept. Throughout this section, we will use the model depicted in figure 1 as a running example of our work and the results can be summarized by the new pre-conceptual schema of figure 3. The rules for ontologically labeling the concepts will be discussed in the following sub-sections.

Note that we are here employing a small fragment of the OntoUML language and of its underlying ontology. For this reason, the pre-conceptual schema presented here should be required to obey all the constraints defined for the OntoUML language. In particular, we use here the following OntoUML constructs: kind, role, property, mode, and relator. The following sub-sections explain the way we can obtain such labels for every concept in the pre-conceptual schema.

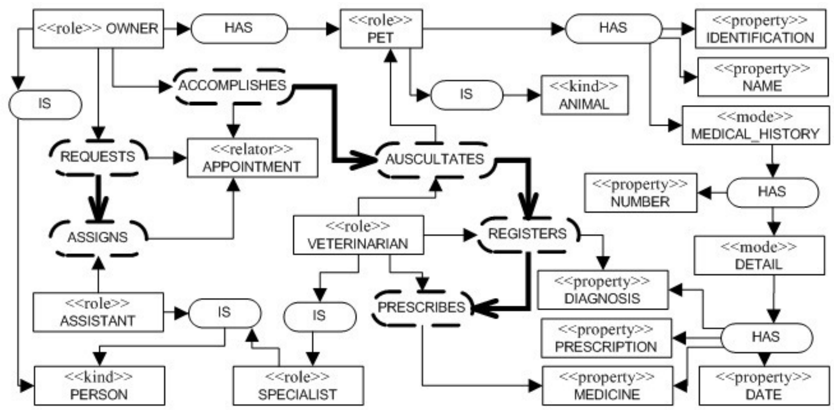

Fig. 3. Pre-conceptual schema of figure 1 after adding ontological labels to concepts.

4.2.1 ⟨⟨property⟩⟩

Guizzardi [8] made a fundamental distinction between a property and the value spaces in which these properties can be projected to. The latter are named *Quality Structures* and are defined as spaces of values endowed with topological and geometrical structures. For instance, associated with the property weight we have a linear quality structure which is isomorphic to the positive half-line of the real numbers. Frequently in conceptual modeling we merely represent the so-called *Attribute Functions* or *Property Functions* which map endurant individuals (objects, modes and relators) into those value spaces. We take here the concepts stereotyped as ⟨⟨property⟩⟩ to represent these property functions and, thus, consider the values they map as their "instances".

Since these entities are functions then must be associated with concepts whose extension forms their domain. Here, we consider a "has" relation between an endurant concept and concept stereotyped as ⟨⟨property⟩⟩ to represent the fact the latter defines a property function for the former. We have then that in a pre-conceptual schema, property functions are recognized by being the end of a "has" relation chain, *i. e.*, a concept which is only in the target end of a "has" relation but never at the source end. It is not the case, however, that a concept in the end of has-relation chain must be a property function, *i. e.*, it can also represent an object concept, a mode concept or a relator concept. Thus, concepts stereotyped as ⟨⟨property⟩⟩ must obey the additional condition of having as instances pure values (points in the aforementioned quality structures) [8].

Using these criteria we can identify in figure 3 the following examples of property functions: "identification" and "name" (as property functions for the type "Pet"), "number (as property functions for the type "Medical History"), and "diagnosis", "medicine", "prescription", and "data" (as property functions for the type "Detail").

4.2.2 ⟨⟨mode⟩⟩

An instance of a ⟨⟨mode⟩⟩ universal is an entity which is existentially dependent on exactly one entity [8]. For this reason, mode concepts are always be in the targeted association end of exactly one "has" structural relationship. In other words, they must be either in the middle or in the end of a "has" relation chain. To put it in a converse manner, if a concept is in the beginning of a "has" relation chain then it should represent an object concept. Notice that, since "has" relations directed at modes are taken here to represent "inherence" relations [8] and, since only modes can "bear" other modes, we have that if a concept (which is not a property function) has a mode concept as antecedent in a "has" relation chain then this concept must also be stereotyped as ⟨⟨mode⟩⟩.

Using these criteria, in our running example we identify the following concepts as modes. Firstly, we can identify that "Medical History" is a mode which is existentially dependent on "Pet" (after all, it represents the medical history *of* the pet). Now, since Detail is not a property function (its instances are not values) and, since it is preceded by a mode in a "has" relation chain, then we must also classify it as a mode concept.

4.2.3 ⟨⟨kind⟩⟩ and ⟨⟨role⟩⟩ (Object Concepts)

Objects are existentially dependent entities and for this reason they are typically found in a pre-conceptual schema in the beginning of a "has" relation chain. In fact, if a concept is found in such a position in a "has" relation chain it should be an object concept. This is not to say that object concepts cannot appear in the middle or in the end of such a relation chain. However, in that case, all antecedent concepts in that chain must also be object concepts. Moreover, in that case, the "has" relation is no longer representing "inherence" relations but ordinary formal or material associations [8].

Object concepts are also typically identified in a pre-conceptual schema by their participation in "IS" relation chains. Moreover, their position in such a chain can also be used to distinguish between different sorts of object types. As discussed in depth in [8], kinds are substance sortals, *i. e.* object universals that supply a principle of identity for their instances. Since principles of identity must be used in every possible situation an entity is referred to, kinds must classify their instances in a so-called *rigid* modality. To be more precise, a type T is rigid if for every instance x of T, x must be instance of T in every possible situation. In contrast, a type T' is anti-rigid if for every instance x of T', there is a possible situation in which x is not an instance of T' [8]. A stereotypical example highlighting this difference is the one between the common sense uses of the terms "person" and "student": while people are necessarily people, students can cease to be students without ceasing to exist. As formally proved in [8], every object must be an instance of exactly one kind which provides the principle of identity that it obeys. Thus, we have the derived constraints that every anti-rigid type must be supertyped by one unique kind, but also that a kind cannot be a supertype of another kind. From this we have both that: (i) if a type is the end of an "IS" relation chain (top-most concept in a taxonomic relation hierarchy) and it is anti-rigid then the model must be deemed incomplete; (ii) if a concept is supertyped by two different kinds then the model must be incorrect.

A final point we should make is regarding object participation in actions. According to authors such as Steimann [16], objects always actively participate in actions by playing roles. For this reason, we take here that object concepts preceding dynamic relationships to be indicative of cases of role.

In the OntoUML fragment considered in this paper we have only two types of object types, namely, kind (rigid) and role (anti-rigid). From this, we have that all object types which are in the middle or in the beginning of an “IS” relation chain must be classified as role concepts. Notice that, following [8], roles are also relationally dependent concepts (e.g., in order to be a Husband, someone must have a wife; in order to be a Student, someone must be enrolled in an Educational Institution). For this reason, typically in pre-conceptual schemas, object concepts directly involved is “has” relations representing formal or material associations are indicative cases of role concepts.

In our running example, using the above criteria, we can identity “Person” as a clear case of a kind. In addition, by checking the constraints underlying the OntoUML theory, we were able to identify problematic cases in the model. Firstly, “Pet” appears in the model as an anti-rigid type without a kind as a supertype. This indicates the absence of concept in the model, namely, the concept “Animal”. Secondly, Veterinarian appears a subtype of both “Person” and “Specialist”, which indicates incorrectness in the model. In this case, Specialist is an anti-rigid concept (a role) and, hence, if we had only an “IS” relation between Veterinarian and Specialist, the resulting model would not satisfy constrain (i) above (neither of the two anti-rigid concepts would have a kind as a supertype). Therefore, we conclude that we should include an “IS” relation between Specialist and Person. In addition, we can remove the direct “IS” relation between Veterinarian and Person given that this relation is now inferred by transitivity from Veterinarian to Specialist to Person.

4.2.4 ⟨⟨relator⟩⟩

After the analysis of our running example reported so far, there is still one concept missing a stereotype, namely, “appointment”. This concept neither participates in “has” nor in “IS” relations. In this case, we must inquiry the stakeholder to discover how this concept’s instances stand with respect to existential dependency. An appointment is a case of a social commitment which is, ontologically speaking, an example of what is named in [8] a relator. A relator is an entity which is existentially dependent on multiple entities. Examples include the marriage of John and Mary being dependent of both John and Mary, a covalent bond between two molecules being dependent on both of them. In the case of our pre-conceptual schema, an appointment is involved in actions in which both an “owner” and an “assistant” participate. This could be taken as an indication that this relator is dependent of both of them (representing an established social bond between these entities). It is important to highlight, nonetheless, that this particular information cannot be derived directly from the model without explicitly consulting the stakeholder.

4.3 Some notes about the translation of ontologically-labeled concepts into conceptual schemas

Ontologically-labeled concepts (running example in figure 3) are the results of applying the described concept categorization to the pre-conceptual schema in this paper. We have two goals in mind for making such categorization: improving the quality of the information represented by pre-conceptual schemas and contributing for a good transformation of pre-conceptual schemas into conceptual schemas.

The first goal is concerned with the completeness and precision of the information we gather from stakeholders. As we discuss in the previous sub-section, we can discover incompleteness and inconsistency in the pre-conceptual schema from the constraints governing the relationships between stereotyped concepts. Also, we can discuss with stakeholders the nature of such stereotypes.

The second goal is concerned with the translation of concepts into UML constructs. For example, we can use the stereotype as a part of the translation into UML classes, leading to more precise UML diagrams. Again, it is important to emphasize that the model in figure 4 is not a valid OntoUML model but merely a UML model using some OntoUML stereotypes. In any case, the enhanced real-world semantics signaled by the stereotypes can be systematically explored to help the modeler choose and justify design choices in an eventual UML design model derived from this one. One example is the use of {readOnly} tagged values combined with a minimum cardinality of 1 to capture the existential dependency relations between modes (or relators) and their bearers [8]. The representation of this lifecycle dependency can then be property handled in eventually derived design and implementation models.

In this paper, we only deal with concept translation and we use the rules defined by Zapata *et al.* [1] for obtaining the rest of the elements (namely operations and associations). Note that consistency is still maintained by the automated translation from the pre-conceptual schema to the class diagram.

The ontological treatment of pre-conceptual schemas can, as we discussed, offer a new way to understand the actual use of them. The work we are presenting in this paper is only the beginning of such understanding, because we need to explore the ontological consequences of the use of both structural and dynamic relationships. As we will note in section 5, there is evidence of mature studies in such an issue, and we need to explore these lines of future work. Also, we envisage to be able to exploit the translation rules defined from pre-conceptual schemas to several conceptual schemas, for instance KAOS goal diagram [9] and OO-Method diagrams [10].

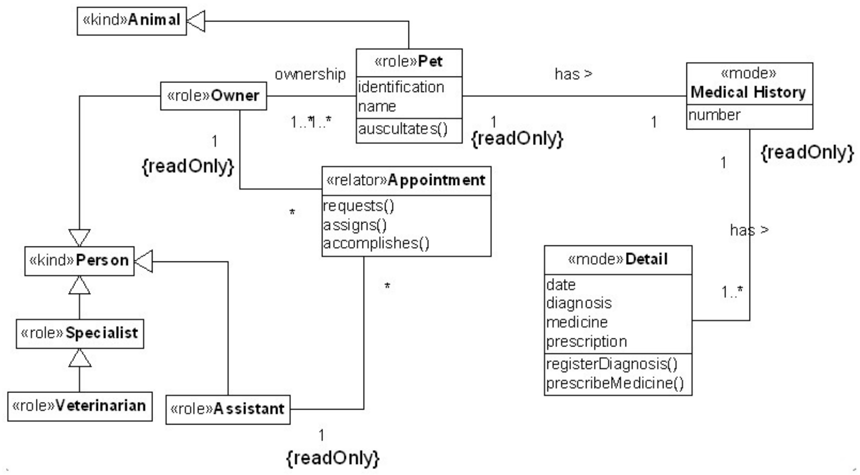

Fig. 4. Resulting class diagram from the pre-conceptual schema of figure 3.

4.4 Remarking comments

A pre-conceptual schema is the result of an agreement between the analyst and the stakeholder. Without concept categorization, the creation of a pre-conceptual schema relies on the analyst expertise. In this way, discussion about the importance of modeling things as concepts has to be led by the analyst, with no available guidelines for this work. As we discussed at the end of section 3, a phrase like "clinic auscultates pet" can be incorporated in the resulting pre-conceptual schema, because before the work of this paper the only constraint to dynamic relationships was the need to be performed by a concept (*i. e.* either a noun or a noun phrase). However, only the most mastered analyst in the pre-conceptual schema usage could discover some misunderstanding in the mentioned phrase.

The approach presented here illustrates an ontological treatment of pre-conceptual schemas providing guidelines for discussing the usage of every concept related to the problem domain. For instance, the current rule "a dynamic relationship must be performed by a concept" can be now translated to "a dynamic relationship must be performed by a concept labeled <<role>>. As a consequence, instead of accepting the fact that "clinic" is a noun and can be the header of a dynamic relationship, now the analyst must be aware about the nature of the concept. In other words, ontological guidelines lead the analyst, in this case, to ask the stakeholder if "clinic" can be considered as a ⟨⟨role⟩⟩. As "clinic" is not a ⟨⟨role⟩⟩ the analyst must ask about the responsibility of some agent to perform the dynamic relationship "auscultates" and the stakeholder can now answer "veterinarian" (which can then be classified as a ⟨⟨role⟩⟩).

In such a similar way, we believe that we can discover the guidelines for every concept category, in order to assign the proper stereotype to every concept inside the pre-conceptual schema.

5 Conclusions and future work

Pre-conceptual schemas are artifacts that analysts and stakeholders may use to discuss and validate information about the problem domain. Despite the natural-language-based validation of pre-conceptual schemas, the concepts inside such schemas are treated in the same way, no matter that they exhibit different behavior. For this reason, in this paper we illustrated an ontologically-based process for assigning ontological labels to concepts, as an effort to categorize the similarities among concepts (and, in this way, provide a real-world semantics to the pre-conceptual schemas). We have demonstrated that concept categorization, based on the work of Guizzardi [8], provides opportunities for checking completeness and precision of the involved concepts.

Some work has still to be done related to this issue:

- The ontological categories employed here are the ones represented in a small fragment of OntoUML. Hence, we can explore other concrete meta-classes in the OntoUML metamodel, as a way to understand the possible implications of incorporating such information in the syntax of pre-conceptual schemas. Particularly, meta-classes like ⟨⟨collective⟩⟩ and ⟨⟨sub-kind⟩⟩ could bring more detailed information about the problem domain.
- We can explore the usage of structural relationships for affecting the concept categorization in a precise way. Guizzardi [8] presents a detailed work about the use of structural relationships as modifiers of concept categorization.
- Extending this study to the context of dynamic relationships.
- Defining guidelines for every concept category, in order to improve the process of characterizing every concept during the analyst-stakeholder interview.
- Exploiting the translation rules of pre-conceptual schemas to other diagrams (for example UML state machine diagram or KAOS goal diagram) and complementing them to bring precision to the translation process.

6 Acknowledgement

This work is partially founded by the COOPEN Scholarship from the Polytechnic University of Valencia and by the Brazilian Research Funding Agencies FAPES (Process Number 45444080/09) and CNPq (Process Number 481906/2009-6 and 309382/2008-4).

7 References

1. Zapata, C.M., Gelbukh, A., Arango, F.: Pre-conceptual Schema: A Conceptual-Graph-Like Knowledge Representation for Requirements Elicitation. Lecture Notes in Computer Sciences, 4293, 17--27 (2006).
2. Zapata, C.M. , Gelbukh, A., Arango, F.: UN-LENCEP: A Controlled Language for Pre-conceptual Schema Specification. In: Memorias de las VII Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento, pp. 269-276. Guayaquil (2008).
3. Novak, M.: Towards understanding Aristotle's categories. Philosophy and Phenomenological Research, 26(1), 117--123 (1965).
4. Roth, W.: Making Classifications (at) Work: Ordering Practices in Science. Social Studies of Science, 35(4), 581--621 (2005)
5. Yang, L., Yang, J., Zheng, N., Cheng, H.: Layered object categorization. In: Proceedings of the 19th Conference on Pattern Recognition, pp. 1--4. IEEE Press, Tampa (2008).
6. Geyken, A., Schrader, N.: Lexikonet - a lexical database based on type and role hierarchies. In: Proceedings of the fifth international conference on Language Resources and Evaluation, LREC 2006. Genoa (2006).
7. Stuckenschmidt, H., Klein, M.: Structured-based partitioning of large concept hierarchies. In Proceedings of the third International Semantic Web Conference, ISWC, pp. 289--303. Hiroshima (2004).
8. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Telematics Instituut Fundamental Research Series, Enschede (2005).
9. Zapata, C. M., Lezcano, L., Tamayo, P.: Validación del método para la obtención automática del diagrama de objetivos desde esquemas preconceptuales. Revista Escuela de Ingeniería de Antioquia (8), 21--35 (2007).
10. Zapata, C. M., Ruiz, L. M., Pastor, O.: Desde Esquemas Preconceptuales hacia OO-Method. Revista Facultad de Ingeniería Universidad de Antioquia, (56), 203—212 (2010).
11. Kop, Ch., Mayr, H. C.: Mapping functional requirements: from natural language to conceptual schemata. In: Proceedings of the 6th IASTED International conference Software Engineering and applications. Cambridge (2002).
12. Mich, L., Garigliano, R.: NL-OOPS: A Requirements Analysis tool based on Natural Language Processing. In: Proceedings of the 3rd International Conference On Data Mining 2002, pp. 321--330, Bologna (2002).
13. Harmain, H., Gaizauskas, R. CM-Builder: An Automated NL-based CASE Tool. In: Proceedings of the fifteenth IEEE International Conference on Automated Software Engineering (ASE'00), Grenoble (2000).
14. Moreno, A., Juristo, N., Van de Riet, R.: Formal justification in object-oriented modelling: A linguistic approach. Data & Knowledge Engineering, 33, 2--47 (2000).
15. Körner, S., Brumm, T.: Improving natural language specifications with ontologies. In: Proceedings of the 21st International Conference on Software Engineering & Knowledge Engineering (SEKE'2009), pp. 552--557, Boston (2009).
16. Steinmann, F.: Role = Interface: a merger of concepts, Journal of Object-Oriented Programming, 14(4), 23--32 (2001).