ONTOLOGY AND METHODOLOGIES FOR INFORMATION TECHNOLOGY: A CASE STUDY FOR COMICS.

Danilo F. Ribeiro
Marcos Luiz Mucheroni

This work is a study of different methodologies for developing ontologies and an ontology is proposed to document the comics. It is made a bibliographic survey to define what are ontologies, in different areas of knowledge; then is made a comparison of ontology development methodologies, which are applied to Comics, respecting their type and characteristics, such as people and organizations that participate in the process of creating and publishing. Some free open source ontology development software were analyzed and compared. As a practical proposal, was outlined an ontology for the branch of Comics, using, for this, the acquired knowledge and tools with which they had contact during the course of work.

Keywords: ontology; methodologies of ontologies; software to ontologies; comics.

1. Introduction

Nowadays is increasing the use of ontologies and folksonomies in digital environments to allow greater ease at search and documentation of various types of digital files. Although the term 'ontology' is originated from philosophy, its current use is very convenient and keeps a lot of similarities with the classic term, meaning "study of being."

According to Lima-Marques (2006, p. 17), "the term Ontology, (…) means the science or study of 'being'. Ontology is the study of the existence of all types of entities, abstract or concrete, which are the world." However, the concept arose long before the term. In ancient times, Aristotle defined it as the study of the being of things. He called this concept of 'First Philosophy' or 'the being as such' (LIMA-MARQUES, 2006, p. 19).

The word 'ontology', as the science of being, emerged in the early seventeenth century. According to Mora (1993, apud LIMA-MARQUES, 2006, p. 33), the first use of the term, definined as 'philosophy of being' was Rudolf Goclenius (1547-1628), in 1613, at his work *Lexicon Philosophicum Wed tamquam clave Philosophiae foresaperintur*, but this concept has not had major repercussions. Thereafter, the term appeared in two papers. The first, titled *Metaphyshica divine, principiis primis eruta, in Entis abstractione repraesentata, ad S,S. Theologicam usum genuinum abusum a hereticum, constans*, was published in 1636 by Abraham Calovius (Calov's - 1612-1686). The second, written by Juan Caramuel of Lobkowitz, was published in 1642 and was called *Rationallis et realis philosophia*. It was Christian Wolff who popularized the term in philosophical circles, by using it in his work *Philosophia prima sive ontologia methodo scientifica pertractata, qua omnes cognitionis humanae principia continentur*, published in 1730 (MORA, 1963).

Regarding the current meaning of the term to Philosophy, Lima-Marques (2006, p. 20) explains that ontology is:

> (...) (LEGRAND, 1986, p. 288): 'the part of philosophy that studies the general categories of existence and its structures in a broader understanding, either from the existence matter or from the essence of the existent matter, applying to the special Metaphysics of several existentialisms... '

To respect this, Mora (1978) said:

> We understand ontology in different ways: on the one hand, is conceived as the science of the being itself, the final or irreducible being, of a first entity in which all other consist, i.e. from which all beings depend. In this case, ontology is truly metaphysics, that is, the science of reality and existence in the proper sense of the word. Furthermore, ontology appears to have as its task the determination of what entities consist and still of what consists being itself. In this case it is a science of essences and not of existences; it is, as stressed lately, theory of objects.

But ontology environments in information technology has specific aspects.

2. Ontologies in digital semantical environment

Over time, the term 'ontology' has been used in other areas of knowledge, beyond philosophy. Because of this, the word has acquired new meanings, more consistent with the areas in which it was used.

In the period between the two World Wars, a Polish mathematician named Stanislaw Lesniewski developed a system he called Ontology. For him, Ontology is a theory and classes and relationships calculus, differing from the propositional calculus (or protothetic) and classes algebra (or mereology), creating an ontological axiomatic (LIMA-MARQUES, 2006, p. 45).

Also according to Lima-Marques (2006, p. 50):

> the formalization of Ontology initiated by Husserl and Lesniewski allowed its use by the areas of Artificial Intelligence, Knowledge Representation[1] and of Information Architecture[2], especially for the developing of Semantic Web (...)

The concept of ontology was first used in the field of computer science, specifically in the area of Artificial Intelligence (AI), in the 1980s. For information systems, what 'exists' is what can be represented. In this context, Gruber (1993, p. 908) defined an ontology as an explicit specification of a conceptualization, a description (like a formal specification of a program) of concepts and relationships that can exist for an agent or a community of agents, a formal statement of the terms, their definitions and axioms that describe their relationships, highly dependent on the universe of discourse and intentional inferences desired.

A conceptualization is an abstract, simplified view of the world it represents for some purpose. This conceptualization is based on a formally represented body of knowledge, covering objects, concepts and other entities related to any area of interest, as well as the relationships between them (GENESERETH AND NILSSON, 1987, apud LIMA-MARQUES, 2006, p. 51).

AI and Web researchers took the term for their own jargon, and for them an ontology is a document or file that formally defines the relations among terms. The most common type of ontology for the Web has a taxonomy and a set of inference rules (Berners-Lee, Hendler, Lassila, 2001).

Ding and Foo (2002, p. 123) say that ontology is a complex multidisciplinary field that is based on knowledge of information organization, natural language processing, information extraction, artificial intelligence, knowledge acquisition and representation.

The World Wide Web Consortium (W3C, 2004) states that an ontology defines the terms used to describe and represent an area of knowledge. Ontologies are used

1 Subfield of AI research that seeks to understand how to represent knowledge; if it is possible to represent anything and how intelligent programs must represent knowledge.

2 Categorization of information into a coherent structure, preferably one that most people can understand quickly.

by people, databases, and applications that need to share information in a domain (a domain is nothing more than a subject or area of knowledge in particular).

The W3C also considers that ontologies need to specify descriptions for the following types of concepts:

- classes ("things" in general) in the various fields of interest;
- Relationships that may exist between classes;
- The properties (or attributes) that classes may have.

Besides, according to W3C, ontologies are generally expressed so that detailed, accurate, consistent, safe and meaningful distinctions can be made between classes, properties and relationships.

Breitman (2005, p. 7) states that "Ontologies are conceptual models that capture and explain the vocabulary used in semantic applications. They serve as a basis for ensuring communication free of ambiguities."

As Lima-Marques said (2006, p. 57):

> For the Semantic Web scholars, ontology consists of a set of terms of knowledge, including vocabulary, semantic interconnections and Logic (simple rules of inference), related to a specific topic, while more complex mechanisms of Logic, such as inferences, are generally considered and treated separated from ontologies.

Reinold and King (2008, p. 8) point out that there are other definitions of ontologies, but all share some simple features: well-defined concepts, relationships and taxonomy.

In the this paper’s context, the adopted notion of ontology is: a model of knowledge representation, consisting of classes (representing concepts), relationships between classes and properties or attributes of classes.

Ontologies can be divided into four types: high-level ontology, domain ontology, task ontology and application ontology. (GUARINO, 1998b, p. 7-8; WELTY e GUARINO[3], 2001, apud YI, 2008, p. 16-17)

High-level ontologies describe very general concepts like space, time, objects, events, actions etc., which are independent of specific areas or issues.

Domain ontologies describe the vocabulary related to a generic domain (medicine, automobiles, comics etc.), specializing concepts defined in a high-level ontology.

Task ontologies describe the vocabulary related to a generic task or activity of high-level ontologies (diagnostic or sales, for example).

Application ontologies are the most specific type of ontologies. They describe concepts depending on a given field and a given task. Their concepts often correspond to roles played by domain entities while performing a certain activity (replaceable unit or component replacement, for example).

Some good reasons for developing ontologies are the possibility to share and reuse knowledge for use in Artificial Intelligence and Information Architecture, beyond the possibility of improvement in web searches, minimizing problems such as semantic ambiguity or polysemy, for example. (LIMA-MARQUES, 2006, p. 52-53)

---

[3] Welty, C. & Guarino, N. (2001). Supporting ontological analysis of taxonomic relationships. *Data & Knowledge Engineering*, 39, 51-74.

Moreover, ontologies define concepts, like a dictionary, establish relationships, like a thesaurus, and categorize concepts, like a taxonomy. In short, ontologies combine elements of these three knowledge structures. You can also define an unlimited number of relationships, represent concepts with phrases instead of words and connect them with various types of relationships at the same time. (KING; REINOLD, 2008, p. 10-11).

Ontologies are tailored for each domain in order to capture its vocabulary along with its unique and expert perspective of information. An ontology can recognize different ways of identifying the same idea. (KING; REINOLD, 2008, p. 12).

For this developed methodologies for ontologies in order to make this work more automated and simple.

3 – Methodologies for ontology development

According to Ding and Foo (2002, p. 125):

> The starting point for creating an ontology can arise from several situations. One can do it from scratch, using only existing ontologies, from a set of information sources, or using a combination of the last two approaches. Various degrees of automation can be used in creating ontologies: from fully manual fashion, passing through half-automated ways to fully automated.

Yet according to the authors, typical methods for creating ontologies are divided into three types: bottom-up, i.e. from more specific to more general concepts; top-down, i.e. from more general to more specific concepts, and middle-out, from the most important concepts, generalizing and specifying when necessary. (DING; FOO, 2002, p. 125).

Some criteria must be taken into account, when creating an ontology. These criteria change according to the characteristics or focus of the ontology. Hwang (1999, apud DING; FOO, 2002, p. 125) proposed some desirable criteria for the ontology created, which must be (1) open and dynamic (algorithmically and structurally, for easy construction and modification), (2) scalable and interoperable, (3) easily maintained (ontologies must have a simple and clean structure, and should also be modular), and (4) independent of context.

Gruber (1993, p. 909-910) establishes for this:

- Clarity: An ontology should effectively communicate the intended meaning of defined terms. Definitions should be objective and independent of social or computational context. All definitions should be documented in natural language.

- Coherence: An ontology should be coherent, that is, should sanction inferences consistent with the definitions. At a minimum, the determinant axioms must be logically consistent. Consistency should also apply to the concepts that are informally defined, like those described in documents and examples in natural language. If a sentence that can be inferred from the axioms contradicts a definition or example given informally, then the ontology is incoherent.

- Extensibility: It should be able to set new terms for special uses based on existing vocabulary in a way that does not require a review of existing definitions.

- Minimal encoding bias: The conceptualization should be specified at the knowledge level without depending on a specific encoding on a symbolic level. A *coding bias* arises when representation choices are made purely for the convenience of notation or implementation.

- Minimum ontological commitment: An ontology should require the minimal ontological commitment sufficient to support the activities of sharing knowledge required. An ontology should make as few demands as possible about the world being modeled, allowing the parties committed to the ontology freedom to specialize and instantiate the ontology as needed. As the **ontological commitment** is based on the consistency and use of vocabulary, it can be minimized by specifying the weakest theory (allowing maximum models) and defining only those terms that are essential to the knowledge communication consistent with that theory.

Guarino (1998a, p. 7) summarized the basic principles of project: (1) clarity about the domain, (2) take the identity seriously, (3) isolating a basic taxonomic structure, and (4) identify roles explicitly.

Uschold and Grüninger (1996, p. 17-18) believe that an ontology must be clear (deffinitions should be as clear and unambiguous as possible), consistent and coherent (an ontology should be internally and externally consistent), extensible and reusable (an ontology should be designed to maximize future reuse and extensibility).

For this paper, the criteria and characteristics that will be considered are: clarity, consistency, extensibility, interoperability, context independence and reusability

Besides the desired properties in ontology, it is also necessary to have a work plan, a method or methodology to achieve the objectives. Descriptions of the following proposals were withdrawn and summarized from the work of Gómez-Pérez, Fernández-López, Corcho (2004, p. 113-148), and King and Reinold (2008, p. 35-50).

3.1 Alguns métodos utilizados

The Cyc method was developed by the U.S. company Microelectronics and Computer Technology Corporation (MCC) in the mid-1980s, when they started to create Cyc, a large common sense knowledge base (KB). The method is divided into three stages, or processes:

- Process I: manual coding of articles and "pieces" of knowledge. Originally handmade, because existing systems did not deal enough with common sense knowledge in order to check for this type of new knowledge. This knowledge was acquired in three ways:

- Coding the knowledge required to understand books and newspapers: seek and represent the common sense knowledge that the writers of the articles assumed their readers already possessed.
- Examination of unbelievable articles: rational analysis of articles with unlikely content as, for example, an article that said that a plane was flying for a year without refueling.
- Identification of issues that "anyone" should be able to respond after reading the text. The KB is increased to be able to answer such questions.

- Process II: coding knowledge aided by tools that use the knowledge already stored in Cyc KB. This process can be performed when tools for analyzing natural language and machine learning tools can use enough common sense knowledge to seek new

common sense knowledge.
- **Process III**: coding knowledge mainly performed by tools that use knowledge already stored in Cyc KB. This process delegates much of the work for tools. To work with Cyc tools, users only recommend to the system knowledge sources to be read and explain the most difficult parts of the text.

Two activities are carried out in all cases:

- Activity 1: development of a knowledge representation and a high-level ontology containing the more abstract concepts.
- Activity 2: representation of knowledge from different fields using such primitive words.

A second method is known as the King and Uschold.

This method was first proposed in 1995, and was extended later (Uschold; Grüninger, 1996). The method's guidelines were proposed based on the experience of developing the Enterprise Ontology, created as part of the Enterprise Project by the Artificial Intelligence Applications Institute at the University of Edinburgh. To create an ontology according to this approach, the following processes must be conducted:

- **Process 1**: Identify the purpose and scope. Clarify why the ontology is being built, its intended uses and the relevant terms in the domain.

- **Process 2**: build the ontology. It is divided into three activities:
- Activity 2.1: capturing the ontology. Made through the following tasks: identifying key concepts and relationships in the field; provide accurate and clear textual definitions for these elements, identify the terms that refer to concepts and relationships and thereby reach a consensus. Textual definitions are made referring to other terms and concepts including notions as class, relationship, etc.
- To identify the concepts in the ontology, Uschold and Grüninger (1996) pointed out the following strategies:
- Bottom-up: first identify the more specific concepts and then generalize them into more abstract concepts. Results in high detail level, however, this approach increases the overall stress, hampers the detection of common aspects of the related terms, and increases the risk of inconsistencies that lead to rework and more effort.
- Top-down: the more abstract concepts are identified first, and then specialized into more specific concepts. The main result is a better control of the detail level, however, start from the top may result in arbitrary choices and impositions, and possible high-level unnecessary categories. As they do not arise naturally, there is a risk of less stability in the model, which leads to greater effort and rework. The emphasis on dividing rather than put together also results in the loss of commonalities inherent in the complex web of interconnected concepts.
- Middle-out: first identify the core of the basic terms, and then specify and generalize as needed. This approach strikes a balance in terms of detail level. Detail emerges only when necessary, through the specification of basic concepts, therefore some efforts are avoided. Starting with the most important concepts, and defining concepts of the higher levels in terms of them, these higher levels categories arise naturally and thus are more likely to be stable. This in turn leads to less rework and less total effort.
- Activity 2.2: coding. This activity involves two tasks: (a) commitment to the basic terms that will be used to specify the ontology of representation (e.g., class, entity

relationship), and (b) writing the code.
- Activity 2.3: integration of existing ontologies. This activity relates to whether and how to use existing ontologies. It can be done in parallel with the previous activities.

- **Process 3**: evaluate. The authors take the definition of Gómez-Pérez, Juristo and Pazos (1995) and claim that it is "to make a technical judgment of the ontologies, their associated software environments, and documentation with respect to a frame of reference (...) (requirements specifications, competence questions, and/or the real world)."

- **Process 4**: document. In this process, the guidelines are established, and possibly differ according to the type and purpose of the ontology. An example of a guideline is to group similar definitions or create naming conventions as: use uppercase or lowercase to name terms, or writing the terms of the representation ontology in capital letters.

During the construction process, the authors propose capturing knowledge, coding it and integrating other ontologies inside the current one. However, according to Uschold and Grüninger (1996), these processes are not only sufficient to have a methodology. You must include a set of techniques, methods and principles for each of the processes, and should indicate the relationship between them (recommended order, interleaving, inputs/outputs).

The main disadvantage of this method is the lack of a conceptualization process prior to development of the ontology. The purpose of a conceptualization process is to provide a domain model less formal than the implementation model, but more formal than the model definition in natural language.

A third method was proposed by Grüninger and Fox and it was published in 1995. It is a formal approach to create and evaluate ontologies.

The methodology is inspired by the development of knowledge-based systems using first-order logic. They propose to identify intuitively the main scenarios, or possible applications in which the ontology is used. Then, a set of questions in natural language, called competence questions, are used to determine the scope of the ontology. These questions and their answers are used to extract key concepts and their properties, relations and formal axioms of the ontology. Knowledge is formally expressed in first-order logic. This is a very formal methodology that takes advantage of the robustness of classical logic and can be used as a guide for transforming informal scenarios in computable models.

The processes of this methodology are:

- **Process 1**: Identify scenarios motivation, i.e. scenarios related to the applications that use ontology. They describe a set of requirements that should be satisfied after being formally implemented. It also derives a set of intuitive solutions possible to their problems. These solutions give a preliminary idea of the informal intended semantics of objects and relations that will later be included in the ontology.

- **Process 2**: prepare informal competence questions, that are questions written in natural language to be answered by the ontology. They play the role of a kind of requirements specification in view of which the ontology can be evaluated. An ontology is not well designed if all competence questions are simple queries, i.e., if the questions cannot be decomposed or composed on more specific or more general ones. They can be divided into more specific questions (or atomic), and the answer to a question can be used to answer more complex questions. Each question is useful as a basis for obtaining assumptions, constraints, input data, etc.

- **Process 3**: specify the terminology using first-order logic. From the questions, is extracted the terminology that will be formally represented by means of concepts, attributes and relations in a first-order logic language. From the answers in natural language, is extracted knowledge to be included in the formal definitions of concepts, relationships, and formal axioms. To create the ontology in first-order logic, designers must perform the tasks of a traditional formalization in first-order logic:
- Identify objects in the universe of discourse.
- Identify predicates. Unary predicates are used to represent concepts, binary predicates for attributes, and binary relations and n-ary predicates to n-ary relations.

- **Process 4**: write competence questions in a formal way using formal terminology.

- **Process 5**: specify axioms using first-order logic. Axioms are defined as first-order sentences using predicates of the ontology. If the proposed axioms are insufficient to represent the formal competency questions and to characterize the solutions of the questions, other axioms or objects should be added.

- **Process 6**: specify theorems of completeness. Once formally established the competence questions, you must define the conditions under which the solutions to the questions are complete. This is the basis of the theorems of completeness to the ontology.

You could say it's a well-founded methodology for the creation and evaluation of ontologies, despite missing some activities of management and support.

A fourth approach known as Kactus was proposed by Bernaras Amaya et al. (1996), in the Esprit Kactus project (Kactus, 1996). One goal of this project was to investigate the feasibility of knowledge reuse in complex technical systems and the role of ontologies to support it (SCHREIBER; WIELINGA; JANSWEIJER, 1995).

This approach is subject to application development. Thus, every time an application is built, the ontology which represents the knowledge necessary for the application is improved. The ontology can be developed reusing other ontologies and it can be later integrated into applications. This approach is divided into the following processes:

- **Process 1**: specification of the application, which provides an application framework and an overview of the components that the application tries to model. In this process, a list of terms and tasks has to be provided.

- **Process 2**: preliminary design based on high-level relevant ontological categories, where the list of terms and tasks developed is used as input for various visions of the global model according to high-level ontological categories, such as concept, relation, attribute, etc. This design process involves searching for already developed ontologies which are then refined and extended to be used in the new application.

- **Process 3**: refinement and structuring of ontology to achieve a final design that follows the principles of modularization and hierarchical organization.

A fifth proposal is METHONTOLOGY, that was developed by the Ontology Group of the Universidad Politécnica de Madrid. It has its roots in key activities identified by the software development process (IEEE 1996) and methodologies in knowledge engineering (Gómez-Pérez et al., 1997; WATERMAN, 1986).

This methodology proposes a lifecycle of ontology development based on evolving prototypes, as this allows you to add, remove and change terms in each new version (prototype).

The conceptualization activity in METHONTOLOGY organizes and converts a informally perceived vision of a domain into a semiformal specification using a set of intermediate representations (IR) based on charts and graphs that can be understood by those involved in the creation of ontology.

The creators of this methodology believe that, to construct an ontology, you must perform the following tasks:

- **Task 1**: create a glossary of terms, which includes all relevant terms of the domain (concepts, instances, attributes, relationships etc.), their descriptions in natural language, and their synonyms and acronyms.
- **Task 2**: create concepts taxonomies. When the glossary contains a considerable number of terms, taxonomies of concepts are created to define the conceptual hierarchy. The METHONTOLOGY proposes the use of four taxonomic relationships defined in the Frame Ontology and the OKBC Ontology: Subclass-Of, Disjoint-Decomposition, Exhaustive-Decomposition and Partition.

A concept C1 is a Subclass-Of another concept C2 if and only if every instance of C1 is also instance of C2. A concept can be subclass of more than one concept in the taxonomy.

A Disjoint-Decomposition of a concept C is a set of subclasses of C instances that do not have instances in common and do not cover C, i.e., there may be instances of the concept C that are not instances of any of the concepts of decomposition.

An Exhaustive-Decomposition of a concept C is a set of subclasses of C which cover C and can have in common instances and subclasses, i.e., there can be no instance of the concept C than is not an instance of at least one of the concepts in the decomposition.

A Partition of a concept C is a set of subclasses of C that do not share instances in common, but that cover C, namely there is no instance of C that is not instance of one of the concepts in the partition.

Before moving on to the specification of new knowledge, one should check if the taxonomy contains no errors.

- **Task 3**: create diagrams of ad hoc binary relations. The goal of this diagram is to establish relationships between concepts. Before proceeding with the specification of new knowledge, we must check if the ad hoc binary diagrams contain no errors, i.e., find out whether the domains and ranges of each argument of each of each relation delimit exactly and precisely the appropriate classes for the relationship. Errors appear when the domains and ranges are inaccurate or over-specified.
- **Task 4**: create the dictionary of concepts. A dictionary of concepts contains all domain concepts, their relationships, their instances, their class attributes and instance attributes. Relations, instance attributes and class attributes are local to concepts, meaning that their names can be repeated in different concepts.
- **Task 5**: detail the ad hoc binary relations. Describe all relations in the dictionary of concepts, and produce a table of ad hoc binary relations. For each relationship, you must specify its name, the names of the source and target concepts, its cardinality, its inverse relation and its mathematical properties.
- **Task 6**: detailing the instance attributes. Describe all instance attributes already included in the dictionary of concepts through an instance attributes table. Each row of the table contains the description of an instance attribute. Instance attributes are those whose values may be different for each instance of the concept. For each

instance attribute, one should specify the following fields: name, the concept it belongs to (attributes are local to concepts), its value type, its unit of measurement, accuracy and range of values (for numeric values); default values, if any; minimum and maximum cardinality; instance attributes, class attributes and constants used to infer attribute values; attributes that can be inferred using this attribute values; formulas or rules to infer attribute values; and references used to define the attribute.
- **Task 7**: detailing the attributes of class. Task analogous to Task 6. Class attributes describe concepts and have their values in classes where they are defined. They are not inherited by subclasses nor by instances. For each class attribute, you must fill out the following information: name, the name of the concept where the attribute is defined, value type, value, unit of measure and value of precision (for numeric values), cardinality, the instance attributes whose values can be extrapolated to the value of this attribute class, etc.
- **Task 8**: detailing the constants. Task analogous to the previous two. For each set, you must specify the following: name, value type (a number, mass, etc.), value, unit of measurement for numeric constants, and attributes that can be inferred using the constant.
-Task 9: define formal axioms. For each formal axiom definition, METHONTOLOGY proposes to specify the following information: name, description in natural language, the logical expression that formally describes the axiom using first-order logic, concepts, attributes and ad hoc relations to which the axiom refers and the variables used.
-Task 10: Define rules. For each rule definition, METHONTOLOGY proposes to include the following information: name, description in natural language, the logical expression that formally describes the rule, concepts, attributes and relations to which the rule refers, and the variables used in the expression.
- **Task 11**: Define instances. Once the conceptual model of the ontology is created, one must define relevant instances that appear in the dictionary of concepts within an instance table. For each instance, you must set: name, the name of the concept to which it belongs, and its attribute values, if known.

It is important to mention that different domain ontologies may have different needs for knowledge representation (KR). The METHONTOLOGY suggests reduce or extend the set of IRs according to the need of the KR in a field and modify the fields of IRs by adding, removing or changing some of the fields previously presented. For example, when creating an ontology only with concepts, attributes and relationships between concepts, it is not necessary to use the IRs that shape formal rules and axioms.

3.2. Comparison of methodologies

The following comparison was drawn from the work of Gómez-Pérez, Fernández-López and Corcho (2004, p. 148-154) and does not include only King and Reinold's method. The authors conclude that there is a great diversity of strategies for developing ontologies. Some approaches consider the application dependent strategy, others semidependent strategy and others the independent strategy. According to them, there are a large variety of strategies to identify the concepts in the taxonomy, while the middle-out approach is the most commonly used.

According to the authors, neither approach covers all the processes involved

in creating ontologies. However, they have established a range of methodologies and methods, presented from the more complete to the less complete:
- METHONTOLOGY is the approach that provides the most accurate descriptions of each activity.
- The methodology On-To-Knowledge describes more activities than the other approaches.
- The strength of Grüninger and Fox's methodology is its high formality degree.
- The Uschold and King method is less detailed than that Grüninger and Fox's
- The Kactus method was used only to create a few applications and ontologies.

The Cyc method is very specific, focused on creating a knowledge base of common sense, and depends on a large amount of texts and articles pertaining to the field.

Most approaches are focused on development activities, especially in the implementation of the ontology, and do not pay much attention to other important aspects related to the management, development and evaluation of ontologies. This is due to the fact that the field of ontology engineering is relatively new.

Almost no approach has a specific tool that gives technological support. Moreover, none of the available tools cover all activities necessary to build ontologies.

The more formal approaches, as well as those that require a large amount of work (to be done not by an individual but by a team) are not considered very appropriate for this study. Because of this, the method of choice is King and Reinold's.

4 – Implementation of an ontology for Comics.

The work was developed in two stages: the analysis of software and choosing the appropriate fields for Comics.

King and Reinold (2008, p. 165) state that: "Tools for building ontologies attempt to simplify the task of creating and using an ontology. Most tools provide some ability to visualize the relationships among concepts and nearly all can generate the ontology into two or more ontology languages".

For them, the best tool to build an ontology is the simplest among those that can handle the work that needs to be done, If the ontology is small, we use a simple visual editor, but If the ontology will change with frequency, along with the current literature, a more complex tool is required. (KING; REINOLD, 2008, p. 166)

The authors state, further, that an ontology language should be evaluated according to the following characteristics: (KING; REINOLD, 2008, p. 158-159)
- Concepts: specifies whether or not the language supports the definition of concepts.
- Relationships: identifies the types of relationships supported by the language. Some languages allow users to define their own relationships, which allows maximum flexibility to support complex semantics.
- Restrictions: indicates whether the language support for the restriction of values or sets specific limits. They ensure that the information makes sense.
- Inference/Rules: indicates whether or not the language supports the ability to infer new facts from existing information.
- Properties/Attributes: describes how the properties or attributes of the elements are expressed in the language.

- Comments: provides various information, including, in many cases, what kind of knowledge structure describes the language better, if the order of elements is significant or not, or special features of the language.

4.1 Analyzing and selecting a software

King and Reinold (2008, p. 168-169) list some basic criteria for evaluating ontology development tools:

- **Usability**: refers to what is simple and intuitive to the user. Look for a tool that derives the basics directly.
- **Ease of navigation and visualization**: To be able to browse an image of the taxonomy is a pleasant aspect; visualization is difficult if the tool tries to show at once large sections of the ontology with all relationships. A better choice would be a visualization tool that allows the user to select specific relationships to display or simply list the concepts within categories.
- **Extraction and learning capabilities**: tools that extract information from documents to help identify concepts of an ontology will be a good choice if your ontology is in continuous development and current literature is readily available. To this point, look for the ability to analyze natural language extraction.
- **Language supported**: there are several languages used to express ontologies. This is still an active area of development and it is important to identify a tool that follow the changes. Choose a tool that export multiple formats in multiple languages, including at least XML, RDF or RDF Schema and OWL.
- **Version control:** Ontologies, by nature, are always an unfinished product. Maintain adequate version control will improve the management the changes of the ontology.
- **Check validation and consistency**: As ontologies grow and become more difficult to view on a single page, increases the chance of inconsistencies and invalid relationships. Some tools come with automatic validation or consistency checking. For example, they will notify you if found cyclical relationships: this occurs when the term X is a subclass of the concept Y which, in turn, is a subclass of X.

The tools analyzed are presented below. For those features evaluated with numbers, the higher the number, the easier the feature (on a scale that goes from 1 to 5). In order to facilitate the work and avoid unnecessary expenses, only free and open source software were chosen, i.e. computer programs whose source code can be made available for use, copy, study and redistribution. The tools discussed include: Amine, CmapTools, Hozo, Neon Toolkit, Protégé and OBO-Edit.

Software / Charactetistics	Amine	Cmap	Hozo	NeOn	OBO-Edit	Protégé
Concepts	yes	yes	yes	yes	yes	yes
Relationship	yes [a]	yes	yes	yes	yes	yes
Restrictions	no	no	yes	yes	no	yes
Inference/Rules	yes	no	yes	yes	no	yes
Atributes	no	no	yes	yes	no	yes
Comments	no	yes	yes	no	yes	yes

Usability	3	5	5	3	3	3
Navigation ease	3	5	5	4	4	4
Visualization ease	4	5	4	4	5	4
Extraction and learning capabilities	yes	yes	no	no	yes	yes
Supported Languages	XML, OWL	N/A	OWL	OWL	OWL	RDF, XML, OWL
Control of version	no	no	yes	yes	yes	yes
Check validation and consistency	no	no	yes	no	yes	yes

a) the creation of relationships and types supported are not intuitive.

Table 1: Software comparison.

The CMAP software proved the easiest to use. However, being a software aimed at creating concept maps, it lacks some of the desirable characteristics, although it could possibly be used for the development of the taxonomy of the ontology.

The Hozo also proved easy to be used, but it has not demonstrated the ability to extract and learning, and is only compatible with the OWL language.

Based on the criteria previously proposed, the Protégé software was chosen.

4.3 Choices of fields for Ontology of Comics

Comics are published in various media and formats. The main types of comics are:

- Periodicals: the most common type of comics. They are published with certain regularity, usually monthly, but there are also special issues, almanacs or commemorative editions. In Brazil, there are several formats, the most common being the so-called "formatinho" smaller and friendlier to children, and the "American format", which is the size of comic books published in the U.S. (VERGUEIRO, 2007, p. 296-299)
- Graphic novels: this is a format closer to the books. They do not have periodicity, being published in single issues, usually with complete stories. They can bring unpublished stories or compilations of previously released material. They usually have a higher cost due to the quality of paper, printing and binding. (VERGUEIRO, 2007, p. 297)
- Series: they are limited editions, which seek to give special treatment to familiar characters, with complete stories, diverse format and better quality paper. Usually have more elaborate script and art, since there are no deadlines like regular series. (VERGUEIRO, 2007, p. 297)
- Comics in newspapers: newspapers were the cradle of comics as we know them, and that's where lots of them continue to be published. In the newspapers are published daily strips and Sunday pages (VERGUEIRO, 2007, p. 298). The strips are characterized by their shape, mostly horizontal, with two to four comics (which may vary) arranged lined in a single column and published generally daily or weekly. Sunday pages are published in the Sunday supplements of some newspapers and,

unlike the strips, usually occupy a full page, which may contain a complete story or a chapter only.
- Fanzines (Fanatic Magazines) are independent publications made by fans (from comics, movies, TV, and other literary fiction). They are produced in different formats with different quality level also (often in mimeograph - at the beginning - or photocopies) and usually reduced editions, not obeying any kind of periodicity or regular publication schedule. It is an area where there is no kind of regulation, preventing the existence of a bibliographic control, however minimal it may be. (VERGUEIRO, 2007, p. 298)
- Digital comics: comics created in or transposed to digital media, which combine traditional elements with visuals, sound and hypermedia elements, such as dynamic diagramming, hypertextuality, animation, among others. (McCLOUD, 2006; VERGUEIRO, 2007, p. 299; EISNER, 2010).

These are the key professionals involved in the creation of a comic book: Writer: person who writes the script; Illustrator: artist responsible for turning the script into drawings that tell the story; Penciler: artist responsible for drawing the next step, when it gives more strength and depth to the original trace using tools such as India ink or brush, Colorist, Cover Artist, Letterer, Translator and Editor.

4.4 The proposed Comics ontology

Chosen the methodology and software to be used, began the work of ontology development. The type of ontology developed was a domain ontology.

The first step was to decide how the ontology would be used. This required answering three questions: who is the audience that uses the ontology; for which purpose will they use it; what types of questions they will do. The audience consists of Comics researchers, comic book readers or anyone who has any curiosity about the subject. They will use the ontology for searching information about characters, creators, genres, stories and publications (in whole or specific issues). The types of questions are relatated to the work of a particular creator (e.g., which stories has he illustrated?), or about a character (in what issue has he/she first appeared?), or on a particular genre (which editions contains stories of a particular genre?), and other questions.

The second step was the creating a list of terms and the third step was the development of a taxonomy from the list. The result is shown in Figure 1 (in Portuguese). The term 'objeto' was inserted as the root of all classes that are part of the ontology, in order to give unity to the taxonomy because, otherwise, we would have only four disjoint classes, and only the class 'quadrinhos' would have subclasses.

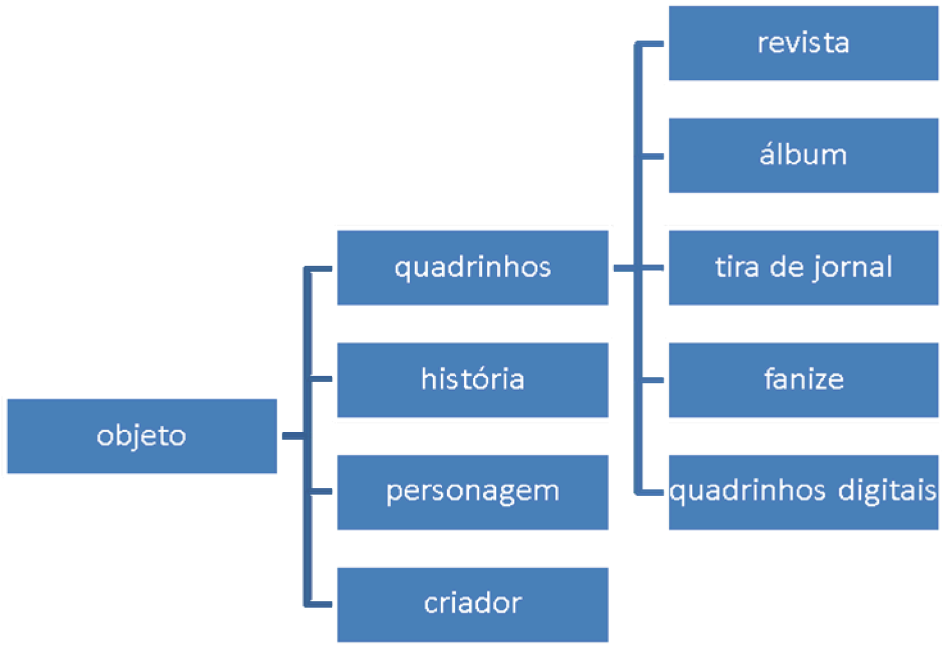

Figure 1 – Initial taxonomy of comics ontology. Source: authors.

The results obtained by the application were placed in Protégé., Placing objects and instance and applying them to practical cases.

4.2. General view of the ontology

Figure 2 shows the general appearance of the ontology developed. It was obtained using the Protégé plug-in OntoGraf, and shows the classes and relationships of the ontology. Classes are represented by rectangles and relationships are represented by lines connecting the rectangles. The solid lines represent the relationships of the type class-subclass. The dashed lines represent other relationships between classes. The class 'Thing' is a mandatory Protégé class (all classes are subclasses of 'Thing').

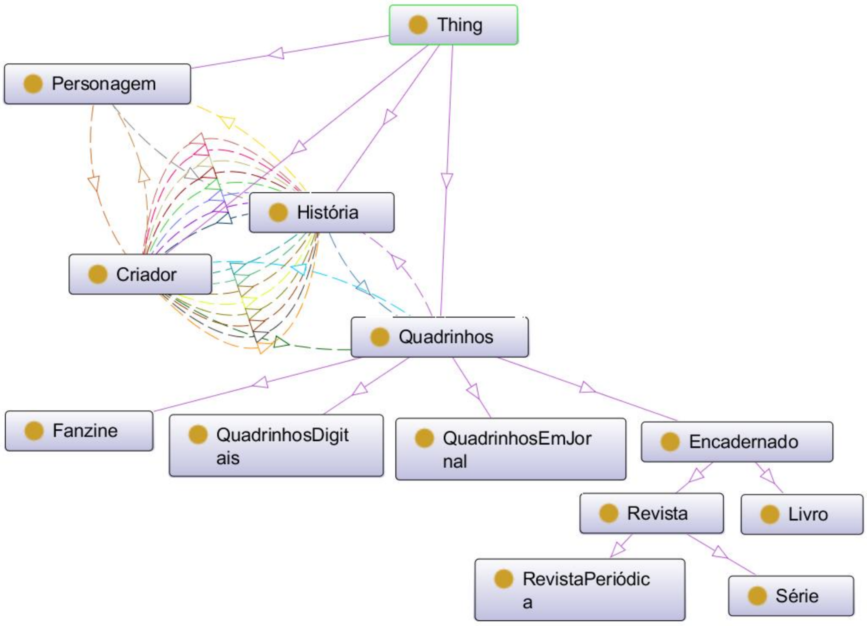

Figure 2 – Classes and relationships in comics ontology. Source: authors.

5. Conclusion

The main objective of this study was to develop a proposal of ontology for the Comics field, in order to permit its use in digital systems for information retrieval. It is possible to state that this goal was achieved.

The bibliographical survey allowed a better understanding of the concept of ontology, to trace a brief history of the term's meaning over time and in different areas of knowledge. Moreover, it was also possible to know, analyze and compare some of the techniques, methods and methodologies for developing existing ontologies, choosing the one we understood was the best in order to achieve the proposed objectives.

We conducted a brief study of the Comics field, in which it was possible to know its main types, characteristics, people and entities involved in the process of creating and publishing a comic strip, and also how all these parts relate to other.

Through analysis of the software development of ontologies, we could know some of the existing tools, their characteristics, their strengths and weaknesses, choosing the most appropriate for the purposes of this project.

REFERENCES

BERNERS-LEE, T., Hendler, J., & Lassila, O. (2001, maio). The semantic WEB. *Scientific American*, 5-7. Recuperado em 03 de junho de 2012, de http://www-sop.inria.fr/acacia/cours/essi2006/Scientific%20American_%20Feature%20Article_%20The%20Semantic%20Web_%20May%202001.pdf>. Acesso em: 03 jun. 2012.

BREITMAN, K. K. (2005). *Web semântica: a internet do futuro*. Rio de Janeiro: LTC.

DING, Y. & FOO, S. Ontology research and development: Part 1 a review of ontology generation. *Journal of Information Science*, 28(2), 126-136. Recuperado em 15 de agosto de 2012, de http://jis.sagepub.com/cgi/content/abstract/28/2/123, 2002.

EISNER, W. *Quadrinhos e arte sequencial: princípios e práticas do lendário cartunista*. 4a. ed. São Paulo: Editora WMF Martins Fontes, 2010.

GÓMEZ-PÉRES, A., Fernández-López, M., & Corcho, O. *Ontological engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. (Advanced Information and Knowledge Processing). Londres: Springer.

GRUBER, T. R. (1991). The role of common ontology in achieving sharable, reusable knowledge bases. In R. Fikes, J. A. Allen, & E. Sandewall (Eds.), *Proceedings of the Second International Conference, Principles of Knowledge Representation and Reasoning*. Recuperado em 25 de setembro de 2012, de http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.1743

GRUBER, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. *International Journal Human-Computer Studies (IJHCS)*, 43. Padova, Itália: Ladsed.

GUARINO, N. Some ontological principles for designing upper level lexical resources. In *Proceedings of the First International Conference on Lexical Resources and Evaluation*. Granada, Espanha. Access in: september 19, 2012, de http://arxiv.org/pdf/cmp-lg/9809002v1.pdf, 1998.
GUARINO, N. Formal ontology and information systems. In FOIS'98. Trento, Itália. Acces in august 27, 2012, de http://www.loa.istc.cnr.it/Papers/FOIS98.pdf, 1998.

KACTUS. (1996). *The KACTUS Booklet version 1.0*. Esprit Project 8145 KACTUS. Access in september 21, 2012, de http://www.swi.psy.uva.nl/projects/NewKACTUS/Reports.html

KING, B.E. & REINOLD, K. *Finding the concept, not just the word: a librarian's guide to ontologies and semantics*. Oxford: Chandos Publishing, 2008.

KOBASHI, N. Y. Fundamentos semânticos e pragmáticos da construção de instrumentos de representação de informação. *DataGramaZero*, 8(6). Acesso em 07 de agosto de 2012, de http://www.dgz.org.br/dez07/F_I_art.htm , 2007.

LIMA-MARQUES, M. *Ontologias: da filosofia à representação do conhecimento*. (Série Ciências da Informação e da Comunicação, 1). Brasília: Thesaurus, 2006

McCLOUD, S. *Reinventando os quadrinhos*. São Paulo: M. Books do Brasil Editora, 2006.

MORA, J. F. On the early history of ‘ontology’. *Philosophy and Phenomenological Research*, 24, 36-47. Recuperado em 05 de agosto de 2012, de http://www.ferratermora.org/essa_ontology.html, september 1963.

MORA, J. F. *Dicionário de Filosofia*. Lisboa: Publicações Dom Quixote. Acesso: em 07 de agosto de 2012, Disponível em: http://www.afoiceeomartelo.com.br/posfsa/Autores/Mora,%20Ferrater/Ferrater%20Mora%20-%20Dicionario%20De%20Filosofia%20(port).PDF, 1978

RIBEIRO, D. F. *Criando uma ontologia para Histórias em Quadrinhos: um estudo prático sobre o desenvolvimento de ontologias.* Trabalho de Conclusão de Curso, Escola de Comunicações e Artes, Universidade de São Paulo, São Paulo, 2012.

USCHOLD, M. & GRÜNINGER, M. Ontologies: principles, methods and applications. *Knowledge Engineering Review*, 11(2), 93-155. Acesso: 22 de setembro de 2012, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.5903&rep=rep1&type=pdf, 1996

VERGUEIRO, W. A linguagem dos quadrinhos: uma “alfabetização” necessária. In A. Rama, & W. Vergueiro. (Orgs.), *Como usar as histórias em quadrinhos na sala de aula* (pp. 31-64). São Paulo: Contexto, 2004.

VERGUEIRO, W. C. S. (2007). Serviços de informação e histórias em quadrinhos. In LARA, M. L. G., FUJINO, A., & NORONHA, D. P. (Orgs.), *Informação e contemporaneidade: perspectivas* (pp. 287-308). Recife: Néctar.

W3C, World Wide Web Consortium. *OWL web ontology language: use cases and requirements*. Acess in july 04, 2012, Av. in http://www.w3.org/TR/webont-req/, 2004.

YI, M. *Topic maps-based ontology and semantic web: ontology-driven information retrieval system*. Saarbrücken, Alemanha: VDM Verlag Dr. Müller, 2008.