doi: 10.15503/emet2017.9.19

# BRIEF QUESTIONS ON THE PHILOSOPHICAL FOUNDATION OF COMPUTER ONTOLOGIES

Francesco Zini
Department of Political Science, University of Siena,
Via P.A. Mattioli, 10 - Piano 0
53100 Siena - Italy
E-mail address: francesco.zini@unisi.it

## Abstract

The paper addresses the role of philosophical ontology as the foundation of computer ontology, which is at the basis of computer language. In particular, the topic of web semantics is addressed to highlight the ever-present risk of evolution in a form of partial reductionism, which relativises the factual reality according to the conceptual scheme that one wants to use/impose.

**Key words:** Computer ontology, philosophical ontology, semantic web, reductionism.

## Computer ontology as a philosophical ontology

On the level of philosophical ontology we can define ontology in very superficial and very general terms as that part of the philosophy that studies "being," so that ontology could be defined as the science of being (of-what-that-is), of the fundamental categories, types and structures of objects, properties and relationships in every sphere of objectified reality[1].

Every field of science naturally has its own ontology, defined by its systematic vocabulary of formulations encoded in its theories. From a certain point of view, as we shall see, in this specific field the ontology is the formation and the determination of a scientific language. In fact, ontologies tend to model themselves on these scientific paradigms, both for the production of scientific theories, but more generally producing theories that represent a progressive clarification of their foundations or justifications. On the level of computer ontology the definition of ontology undergoes a first fundamental reduction; the main objective of ontology is not to answer generic questions about ontologies: what are the different types

---

[1] The term ontology derives from the Greek *òntos* and from *lògos* ("discourse") and is often used in such a way as to be synonymous with the philosophy of being, but in reality the discourse on being has had multiple substantivisations making the being ens, essentia, existentia, being (*Seyn*), being essente (*Seiendheit*), essente, being there, being-being and letting-being, being-for, they-with, being. From a certain point of view the whole Western philosophy is summarized by a discourse on being: certainly Plato, Aristotle, Hegel and Heidegger have used the determination of the verb "to be" to express their philosophies in very controversial terms but simply (and reductively) with ontology we have tried to classify entities both objectively and subjectively understood.

of ontologies? What is the purpose of using ontologies in an application? What methods can I use to construct an ontology? There are different types of ontologies, but indicating a meaning, for example, of the term "computer ontology," can be misleading because it indicates different objects depending on the context. For example, an ontology can be a dictionary, a thesaurus, a logical theory in the field of information retrieval, or a model represented in the data field or a schema in the context of databases.

But what is the specificity of computer ontology? Computer ontology is configured as a specification of a conceptualisation that concerns the very meaning of ontology. The term "computer ontology" seems to generate a series of discussions on the etymological meaning. If, on the one hand, we observe that ontology is a discourse on being or a study of being, from ντος (*òntos*) = being + λόγος (*logos*) = speech, on the other hand, the difficulty of precisely defining "what exists," it relates to the same story on the fundamental (or metaphysical) questions of philosophical ontology, as Descartes recalls in the metaphor of the tree: «So all philosophy is like a tree, whose roots are metaphysics, the trunk is the physics and the branches that arise from this trunk are all the other sciences, which are reduced to three main ones, that is medicine, mechanics and morals: I mean the highest and most perfect morality, which presupposing an entire knowledge of the other sciences is the last degree of wisdom. Now, as it is not from the roots, nor from the trunk of the trees that are harvested the fruits, but only from the ends of their branches, so the main utility of philosophy depends on those of its parts, which can not be learned except for last» (Cartesio, 2009, pp. 15-16)[2].

So any attempt to define "what exists" will lead to a classification of objects (more precisely of the entities) that make up the world (or the real). Therefore we can in our case add a definition of ontology that will be useful for the following definition of computer ontology: it is the rigorous, methodical, orderly description of a world or a representation. This descriptive characteristic or "descriptive" approach is the element from which computer ontology originates and the application of ontology to computer science. The other classic example commonly used in reference to the problem of the definition of ontology is the famous painting by Magritte which represents a pipe with extreme realism. Under the picture it is written: "Ceci n'est pas une pipe," "This is not a pipe." Certainly the realist representation of a pipe (however faithful) is not a pipe, but just as certainly the painting faithfully represents a pipe and also the picture exists. Then some objects (pipes) exist at one level and others exist as representations of other objects as paintings. This same ontology will be the one which deals with the study of forms of legal reasoning, the logic and the theory of argumentation tend to focus on a deductive reconstruction to justify a decision, ignoring the dialectical process that leads to justification. Recently, the developments in Artificial Intelligence and Law have paved the way to overcome the separation between deductive and non-deductive arguments (Gruber, 1993, p. 199; Guarino, 1998, p. 625).

On the one hand, it is important to distinguish these different forms of computer ontologies to clarify their contents, their use and their objective; on the

---

[2] Translated by the author.

other hand, it is also necessary to define precisely the vocabulary derived from the term computer ontology. If we try to clarify some aspects of the ontological question starting from the meaning to be attributed to ontology, we see in the field of information processing information systems collect the data they want to collect because they have their own terms “idiosyncratic concepts” that represent the information received. In attempting to put this information together, methods for resolving terminological and conceptual incompatibilities must be found. Initially, these incompatibilities were solved case by case. Little by little, it became clear that the once-for-all provision of a “taxonomy” was to constitute the “common backbone” to the objective entities relevant to an application domain that offers significant advantages over the case-by-case resolution of incompatibility. This taxonomy constitutes a “common backbone” and is indicated by informatics or better said to be “information scientists” as an ontology.

## COMPUTER ONTOLOGIES AS KNOWLEDGE SYSTEMS

Computer ontologies also contain many definitions that we can define as binding. Ontology is an explicit formal description of a discourse made up of classes or concepts in which the properties of each concept describe the various characteristics and attributes of the concept (property). Thus, computer ontology indicates a logical-artificial linguistic system that was designed for a purposea of allowing a model of knowledge of some real or imagined “domain,” making it communicable. From this point of view the term ontology in the context of information technology and information sciences refers to the creation of an informatics language.

As far as the purposes and characteristics are concerned, what is relevant for computer ontology lies in its instrumental finalistic character: an ontology is always for something or someone. The design of ontologies always serves something and remains “functional” in order to allow the sharing of knowledge and reuse. In this context, ontology is a specific set of definitions of a formal vocabulary. Although this is not the only way to specify an ontological conceptualisation, which has some properties for sharing knowledge, for example between software. Ontological semantics is configured as an agreement to use a vocabulary (that is to say query-instance and/or make statements) in a coherent (but never complete) way with respect to the theory specified by an ontology. So the design of ontologies takes place so you can share the knowledge with these agent-users who will then use them and we can define ontology as a set of basic representation with which to model a domain of knowledge or discourse. The definitions of basic or primitive representations include information about their meaning and the constraints on their logically coherent application. In the context of database systems, ontology can be seen as a level of abstraction of data models, analogous to hierarchical and relational models, but intended for modeling knowledge about individuals, their attributes and their relationships with other individuals (Gruber, 2009).

Therefore ontologies are generally specified in languages that allow an abstraction starting from aggregations of data and strategies for the implementation of the application. In practice, the languages of ontologies (which are very similar to the expressive capacity of linguistic logic) are used to model databases. For this reason, ontologies are placed at a "semantic level," while the database schema are data models at a mathematical level. Thanks to their independence from the lowest level data models, ontologies are used to integrate heterogeneous databases, allowing interoperability between different systems by specifying interfaces and services based on independent knowledge. For example, in the *Semantic Web* standard ontologies are considered "prerequisites" because there are standard languages and a variety of commercial and open source tools for creating and using databases (Lubyte &Tessaris, 2007, p. 387).

The objects of computer ontology are themselves "determined" by language and exist only on the basis of historical and social conventions: their meaning varies with social contexts, historical periods, levels of discourse. The role of computer ontology is to describe such objects by making meaningful assumptions in terms of (meta) minimum properties that can be universally shared. In a more restricted (or more technical) sense, with computer ontology, a shared meaning of concepts is defined, to facilitate communication interchange, network inter-activity, the reuse of lexical resources, especially on the harmonization of contents[3].

## Computer ontology as conceptualisation and abstraction

In the context of knowledge sharing the term ontology also serves to indicate a specific conceptualization. An ontology is a description (as a specific formal program) of concepts and relationships that may exist for an agent or community of agents and this definition is consistent with the use of ontology as a set of concepts.

In this regard, ontologies operate as a "disciplinary mechanism": an instrument for regulating information. A formally represented body of knowledge that is based on a conceptualisation of objects, concepts and other entities that are presumed to exist in some area of interest with the relationships they maintain between them. So a conceptualisation in computer ontology is an abstraction of a simplified view of the world that we want to represent for some purpose. Every system of knowledge commits itself to some conceptualisation, explicitly or implicitly, and an ontology of this type is an explicit specification of a conceptualisation; but the term "computer ontology" always refers to a "systematized" system in which the term "exists" is defined by what can be represented. When the knowledge of a domain is represented in such declarative formalism, the set of objects

---

[3] In the opposite direction there goes, for example, the social ontology of J. Searle (2006). *La costruzione della realtà sociale* [The construction of social reality]. Torino: Einaudi, according to which the role of ontology is to construct a social ontology or a political ontology, describing the nature, the properties, the role of social entities. These objects, such as nations, social classes, communities, associations, governments, banks, universities, but also rights, obligations, powers, money, copyrights, patents, do not exist per se, nor a physical identity, but populate social life and are the subject of any discourse on politics, social behaviour, justice as a feature that allows them to be identified.

that can be represented can become a universe of discourse which we could also define as an ontological horizon.

This set of objects and relationships that can be described among them is reflected in the vocabulary of the formal representation with which a program is based on the knowledge represented. Thus, in the context of computer ontologies, we can describe the ontology of a program by defining a set of terms that represent concepts. In this computer ontology it is used to associate the definitions of the names of the entities to the universe of the discourse (for example, classes, relations, functions, or other objects) with a readable text that describes what the names mean, the formal axioms that limit the interpretation and the good use of these terms. Therefore formally, ontology is the declaration of a logical theory in which computer ontologies commonly serve to describe ontological commitments for a set of agent-users, so that they can communicate with a discourse without necessarily operating on a globally shared theory. One could say that an agent-user agrees to use an ontology if his or her observable actions are consistent with ontological definitions. The "actions of agent-user-clients" - including knowledge base servers and knowledge-based systems - can be viewed through a functional interface, in which a user-client interacts with an agent to make logical or required statements.

The level of knowledge is a level of description of the knowledge of an agent-user that is independent of the representation at the symbolic level used internally by the agent. Knowledge is attributed to user agents based on the observation of their actions; an agent "knows something," if he acts as if he has the information and acts rationally to achieve his goals. Therefore pragmatically, an ontology defines the "common vocabulary" with which questions and affirmations are exchanged between agents, who agree on how to use the shared vocabulary in a coherent and constant way. User agents who share a vocabulary do not necessarily share a knowledge base, so each user-agent commits himself to using an ontology to answer all the questions that can be formulated within the common vocabulary.

So the commitment to a common ontology is a guarantee of coherence, but not of completeness: there will always be new questions and affirmations that use the vocabulary defined by the ontology. Although ontologies are often equated with taxonomic class hierarchies, the class definition is the subsumption relation between these forms, which introduce only the terminology and do not add any knowledge of the world.

To specify a conceptualisation, it is necessary to assert axioms that constrain the possible interpretations of the defined terms. So computer ontology arises from the effort to create standards of inter-operability of a technology to make systems of knowledge comprehensible. In this context ontology is defined as an "explicit specification of a conceptualisation," which are in turn the objects, the concepts, and the other entities that are supposed to "exist" in some area of interest and the relationships they hold between of them. As already noted, the essential points of this definition of ontology are enclosed in the same definition according to which an ontology indicates the concepts, relationships and other distinctions that are relevant for modeling a domain. Vocabulary definitions represent classes, relationships that provide meanings for vocabulary and formal constraints on its consistent use.

One objection to this definition is that it is too broad, allowing a range of specifications of simple glossaries (thesauri) and logical theories compiled in predicate computation.

But this applies to data models of any complexity; for example, a relational database of writing codes as words in which a single table or column is still an instance of the relational data model. On the other hand, the W3C semantic Web standard currently in use suggests a formalism that specifies ontologies that encodes in different variants that vary in expressive power. This reflects the intent that an ontology is a specific form of an abstract data model (domain conceptualisation), which is independent of its particular form. Hence the applied ontology in the context of a software or a database has a specific vocabulary to make statements, providing a language to communicate with the user agent. If a user-agent claims that this interface is no longer needed to use the terms of the ontology as an internal encoding of his or her knowledge, the definitions and formal constraints of ontology do not place restrictions on what can be understood. In essence, those who participate in this ontology are committed to using the vocabulary of ontology (Gruber, 1993, p. 199).

Similarly, while an ontology must be formulated in some representation languages, it is intended to be at the semantic level, which indicates a strategy independent of a modeling of data such as a database. For example, a conventional database model can represent the identity of individuals using a primary key that assigns a unique identifier to each individual. The design of ontologies deals with making representational choices that "capture" the distinctions of a domain at the highest level of abstraction, using data. The patrimony of a computational ontology with respect to philosophical ontology is like a rich body of theories on how to make ontological distinctions in a systematic and coherent way. For example, many of the insights of "formal ontology" motivated by understanding "the real world" can be applied when building computational ontologies for the database world. When ontologies are codified in standard formalisms, it is also possible to reuse ontologies previously designed for different motivations of knowledge or human language. In this context, ontologies embody the results of academic research and offer an operational method for putting theory into practice in database systems, being as clear as possible about the meanings of terms (Gruber, 1995, p. 907). The "primary key" identifier is a formal construction of the design process and does not denote "something" in the domain. For example, in the formalisms typical of ontology one might be able to say that an individual was a member of the class or with some attribute or value without reference to any implementation models, such as the use of primary key identifiers (Guarino, 1998, p. 625).

Ontologies are generally formulated in languages that are closer to the expressive capacity of logical formalisms. This allows the ontology designer to be able to affirm semantic constraints without imposing a particular coding strategy.

## COMPUTER ONTOLOGIES SUCH AS SHARING DATA COMMUNICATION

Why would anyone want to develop an ontology? Some of the reasons are to share "in common" the understanding of the information structure with other people or software agents or to allow the reuse of domain knowledge or to separate domain knowledge from operational knowledge. Sharing and common understanding of the information structure between people or software agents is one of the most common goals for developing ontologies. For example, suppose several different websites contain medical information or provide doctors with e-commerce services. If these websites share and publish the same basic ontology of terms that everyone uses, user agents are able to extract and aggregate information from these different sites. Agents can use this aggregated information to answer user questions or as input data from other applications. Being able to re-use domain knowledge has been one of the driving forces of the recent surge in ontology research, such as models for many different domains that must represent the notion of time. This representation includes the notions of time intervals, time points, time measurements, and so on. If a group of researchers develop such an ontology in detail, others can simply reuse it for their own domains. Furthermore, if we need to build a large ontology, we can integrate several existing ontologies that describe parts of the "great domain." Often the development of an ontology is similar to the definition of a data series and their structure for other programs to be used. Problem-solving methods, independent network applications and agent--users that use ontologies on the basis of knowledge are built by data ontologies.

We have already observed how ontologies are part of the World Wide Web Consortium (W3C) standards for the Semantic Web, where they are used to specify conceptual vocabularies in which the exchange of data between systems, serves to provide services, to answer questions, to publish reusable "knowledge bases," and offer services to facilitate interoperability between multiple heterogeneous systems and databases. The key role of ontologies with respect to database systems is to specify a c.d. representation of data modeling at a level of abstraction that lies above specific databases (logical or physical); so that data can be "exported," translated, interrogated and unified between systems and services developed independently. The main successful applications to date include database interoperability, database search and web services integration.

In particular on the ontological level in the semantic web we can try to give an answer on the "thing" we mean by computer ontology and in particular on "what" ontology, since we see that computer ontology is a kind of controlled vocabulary of well-defined terms with a specific meaning with relationships between these terms, able to be interpreted by both users to process data. Concepts, instances and properties refer to one or more symbols. The symbols are terms that can be quickly understood from their reading. And finally all these "ontological components" are connected through relationships. Semantic relationships link only concepts of set (for example, the relationship indicating the position of the concept of "city" is localized in a concept of "country"), others are relationships to connect, others are just semantic relationships because some relationships can be contextual and cannot be generalized to all statements of their concept. An example of a relationship is the term city called Paris, which is located in the country called France. All cities are located in a country. A contextual relationship can be unraveled in the term person named "John Travolta" which is located in the term "city named" Paris "at the moment January 31, 2010." Terminology relationships to express relationships can have: for example, the term "person" as a synonym for the term "human being" (Chimaera, 2000).

According to the use of these components, we distinguish the types of ontologies that we will not analyze, but each ontology explains that the type of language is normally used to define the ontology according to the aims to be achieved. The same classifications use formal languages, but the assumption is that natural language is in itself ambiguous and vague; therefore the vocabulary that is used is always checked for the reasons and meanings that determine the individual definitions: the ontology establishes the boundary, the conventional system of signs and languages giving a specific meaning above all to create an information technology that can be reused in those terms which determine a level of compatibility between ontological concepts and between theories of the domain they transmit. For example, medical ontology is a fundamental model for the knowledge and communication of ontologies. The presence of additional knowledge is illustrated and some problems in the creation and alignment of biomedical ontologies are discussed (Keet, 2003). The ontological techniques have been widely applied to the physician and biological research. The most effective example is the GO Project, which is an important bioinformatics initiative aiming to standardize gene representation and gene product attributes between species and databases. GO provides a controlled vocabulary of terms to describe gene product characteristics and gene product annotation data, as well as tools for accessing and processing data (Keet, 2007, pp. 65-67).

This ontological research on the level of computer language can be defined as a computational model that involves some portion of domain (of the world). The model describes the semantics of the terms used in the domain: the term "ontology" often refers to a semantic network. The network is intended as a graph in which nodes are concepts or individual objects whose architecture is represented by relationships or associations between concepts. Network semantics always increase: we have moved from properties and attributes to constraints, functions and rules, which govern the behaviour of concepts. This ontology consists of a finite set of concepts, as a set of properties and relationships of these concepts within the structure of a graphic scheme. From this derives the heterogeneity of the different ontologies developed by the different language systems as an intrinsic characteristic of the computer ontologies developed by different subjects for the same domain. In this case systems can occur in which the semantics are heterogeneous in different ways.

Several ontologies may use different terminologies to describe the same conceptual model, which in different terms could be used for the same concept: an identical term could be adopted for different concepts. Even though two ontologies use the same name as a concept, the associated properties and relationships with other concepts are more likely to be different. For example, Ontology Matching stands for an ontological correspondence of an Ontology Mapping scheme. The process of determining correspondences is implemented between the concepts of heterogeneous ontologies, often designed by subjects distributed in different territories. Machine Learning is a scientific discipline that deals with the design and development of algorithms that allow computers to change behaviour based on available data. One of the main objectives of the machine learning research is to learn how to automatically recognize complex schemas and make intelligent decisions based on data (http://www.ontologymatching.org).

Finally, there is not a single correct ontology-design methodology for which the "ontological ideas" that arise are the most useful ones in the ontology development experience. One of the most difficult decisions to make when designing an ontology is when to introduce a new class or when to make a distinction across different property values. It is difficult to navigate in an extremely "ramified" hierarchy with many foreign classes and a hierarchy that has too few classes with too much coded information. Finding the right balance is not easy because there are several practical rules that help you decide when to introduce new classes into a hierarchy. In other words, we usually introduce a new class into the hierarchy only when there is something we can insert in this class, which can not be placed in a different class (higher or lower). However, sometimes it can be useful not to create new classes, to introduce some new properties within a class. For the hierarchical classes, it is better not to introduce new properties.

For example, some ontologies include large reference hierarchies of common terms used in the domain, such as an ontology of a medical system submitted to data recording that may include a classification of various diseases. The classification can only be a hierarchy of terms without properties or with the same set of properties. In this case, it is still useful to organize the terms of a hierarchy rather than a simple list because then it will be possible to allow easy exploration and navigation so as to allow a doctor to easily choose a level of generality of the term that is appropriate for the situation.

Another reason to introduce new classes without new properties is to model concepts among which domain experts commonly make a distinction, since we use ontologies to facilitate communication between industry experts and system-based systems experts knowledge. If a distinction is important in the field and we think of objects with different values for distinction, like different types of objects, then we should create a new class for that distinction. Considering the potential individual cases of a class can also be useful for deciding whether to introduce a new class or not. Usually when using an extrinsic property rather than the intrinsic properties of concepts to differentiate classes, the instances of these classes will often have to migrate from one class to another, while usually numbers, colours, and positions are values that do not; they cause the creation of new ones: in fact to think of the difference between an instance or a class means deciding whether a particular concept is a class of an ontology or a single instance, and deciding where the classes of individual cases begin depends on what the potential applications of ontology are. Are the individual instances the more specific concepts represented in a knowledge base or are they still concepts that form a natural hierarchy, which we should represent as classes within a new formal ontology? (Floridi, 2003, pp. 155-6; Maronaldo, 1990)

In this scenario, the development of applications based on the semantic web will be opened not only as open access, but as an interaction of information and languages with bodies (see the cyborg-user or the Internet of things) that will require an informatic background. In the near future, models will be created in visual environments and the computer models will be given by the same applications, with continuous self-documentation and performances always work in progress and open access. As in the case of Artificial Intelligence (AI), it will not be the only users, but the computers write the code of the applications where the greatest functionality will be created reusing and combining pre-coded code functionality: in this sense, all the application software will be a huge network of ontologies.

### Computer ontology and philosophical reductionism

Concluding, the substantial logical assumptions of the semantic web are not so much in the applied language, but in philosophical-theoretical questions that underlie linguistic and semantic choices that have implications on the philosophical and juridical plane as well. We have observed how ontology is certainly a philosophical category that cuts and "reduces by deciding" the objective and subjective qualification of semantic language: signs are meanings and meanings are preliminary signifiers. For this reason the choices of semantic organization in which to insert the terms of reference are included in the examination of the analysis of ordinary and legal language[4].

If the philosophical ontologies have developed in every field of study of science such as law, computer science, medicine, engineering, the reason lies in the ability of these ontologies to apply the tools of the ontology in order to solve the problems of linguistic-communicative comprehension that arise in these areas: «We consider it as essence or existence, we consider it as a copula position of existence, the being of the being does not belong to the field of preaching, because it is already involved in every preaching in general and makes it possible».[5]

As regards the philosophical-juridical ontologies in this sense, it is true that we can distinguish the meta-norms (the rules on interpretation, the rules on conflict resolution, the analogies, the applicative norms) that serve those who must apply the right; but then we can also distinguish the ontologies that concern the relations between the different legal sources by competence or according to a hierarchical

---

[4] In Core Legal Ontology, law is seen as the description of the ideal way in which things should take place in the world (situations): obviously not all behaviour affects the world of norms, not all norms have to do with real situations. Core Ontology is developed on the foundational ontology (http://www.estrellaproject.org/lkif-core/). See also DOLCE, a project of the Laboratory of Applied Ontology of the Institute for Cognitive Sciences and Technologies of the C.N.R: of Rome, which collaborates with ITTIG on the development of legal ontology.

[5] My translation of Derrida J. (1971), *La scrittura e la differenza.* Torino: Einaudi, p. 172: «Lo si consideri come essenza o esistenza, lo si consideri come copula posizione di esistenza, l'essere dell'essente non appartiene al campo della predicazione, perché è già implicato in ogni predicazione in generale e la rende possibile».

criterion. Thus the prediction of a hierarchy of norms as of ontological classes, as we have observed for the construction of a common semantics, appears to be ineluctable.

As noted in the choice of the language of terms, boundaries "you always decide" what to include and what to exclude, what to insert and what do not consider pertinent: you build an open system, but the degree of openness "is already" a cut, a cut or an include and a "paste." As such, in including a plurality of heterogeneous data, in not considering them superimposable, in considering them on the same logical-egalitarian level, we already identify ontologies "in a strong sense" with a system of rules and exclusions. This computerized ontological reductionism already present as a feature in many philosophical and juridical ontologies is therefore also recognisable in the computer science, which becomes and remains for this reason a philosophical ontology.

## REFERENCES

[1] Cartesio (2009). *Opere Filosofiche, vol. III* [Philosophical Works]. Roma-Bari: Laterza.

[2] Chimaera (2000). Chimaera Ontology Environment. Retrieved from: (www.ksl.stanford.edu/software/chimaera).

[3] Floridi L. (Ed.) (2003). *Blackwell Guide to the Philosophy of Computing and Information.* Oxford: Blackwell.

[4] Gruber T. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition, 5 (2)*, 199-220.

[5] Gruber, T. R., (1995). Toward Principles for the Design of the Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies, 43 (5-6)*, 907-928.

[6] Gruber T. (2009). *Encyclopedia of Database Systems.* In: L. Liu, & M. T. Özsu (Eds.), (pp. 1963-1965). Springer: Verlag.

[7] Guarino, N. (Ed.). (1998). *Formal Ontology in Information Systems.* Amsterdam: IOS Press.

[8] Keet, C.M. (2003). Biological data and conceptual modeling methods. *Journal of Conceptual Modeling,* 29.10.2003.

[9] Keet, C.M., & Rodrıguez, M. (2007). Toward using biomedical ontologies: trade-offs between ontology languages. *Proceedings Of Semantic eScience, Vol. WS-07-11 of AAAI,* 65-68.

[10] Lubyte, L., & Tessaris, S. (2007). *Extracting ontologies from relational databases.* Proceedings of the 20th International Workshop on Description Logics, 387-395.