Chapter 3
Introduction to Ontology

3.1 Introduction

It is important for agents to communicate and interact with each other, especially if they are part of the same multi-agent system. In most cases, different agents are working collaboratively towards the same goal. They need to talk to each other, share tasks, exchange results etc. Here, it is important that agents understand each other; for example, they need to speak the same language or be able to translate and understand the language spoken by other agents.

Ontologies are used to establish effective communication between different agents. Ontologies specify the terms used in agents' communication and provide the exact meaning of those terms relative to other ontology terms and within a specific context. Ontologies provide the agent with the domain knowledge and enable it to function intelligently.

In this chapter, we will introduce ontologies. We will provide a definition of ontology and explain associated terminology such as ontology commitments, ontology representation, ontology classification; we will give a formal description of ontologies and ontology design criteria.

3.2 Ontology Origins

The term "ontology" has its origins in metaphysics and the philosophical sciences. Aristotle first introduced ontology as a philosophical discipline that investigates existence or being (Corazzon 2000). Ontology explains the nature and essential properties and relations between all beings. Ontology is based on the truth; its nature being independent of one's background, present understanding, perspective and knowledge of the world.

Artificial Intelligence researchers first borrowed the concept of ontology from Philosophy (Russel and Norvig 1995) (Figure 3.1). Since then, the notion of ontology has become a matter of interest to information and computer scientists in general. In information and computer science literature, the term "ontology" takes on a new meaning that is partly related to its philosophical counterpart.

Usually, a team of researchers who need to share information in a specific domain propose an ontology. The researchers may be inspired to design a common ontology for various reasons (Gruber 1995) such as:

- common understanding of the domain knowledge (knowledge within a community of people committed to a common goal)

M. Hadzic et al.: Ontology-Based Multi-Agent Systems, SCI 219, pp. 37–60.
springerlink.com

Fig. 3.1 Ontology originates from Philosophy and Metaphysics

- sharing of the domain knowledge
- reusing of the domain knowledge
- analyzing domain knowledge
- separation of domain knowledge from operational knowledge
- making domain assumptions explicit

3.3 Ontology Definition

Different ontology definitions can be found in the computer and information science literature. But all researchers agree on the importance of ontology in the representation, sharing and reuse of existing domain knowledge (Gómez–Pérez 1998).
The most prevalent ontology definition is the following (Gruber 1993).

> "*A body of formally represented knowledge is based on a conceptualization… A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly. An ontology is an explicit specification of a conceptualization.*"

We analyze Gruber's ontology definition in more detail. Three main concepts are central to this definition: "formal", "domain conceptualization" and "explicit". "Formal" refers to knowledge representation that is mathematically described and machine readable. A "domain conceptualization" is an abstract model of a phenomenon, i.e. an abstract view of domain concepts and relationships among them. "Explicit" expresses clear and precise definitions of concepts and their relationships.

Gruber puts further emphasis on ontology sharing. There exists an agreement between ontology users that the ontology represents consensual knowledge. This knowledge is not related to an individual, but is accepted by a group (Fensel 2001).

In the same article, Gruber states further:

> "*the set of objects that can be represented is called the universe of discourse… definitions associate the names of entities in the universe of*

discourse (e.g., classes, relations, functions, or other objects) with human-readable text"

Here, terms "objects" and "entities" are mentioned within an ontology context. This can be quite confusing for people who are being introduced to the concept of ontology for the first time and have a background in entity-relationship and object-oriented modelling. In this book, we will avoid using terms associated with other modelling techniques.

We adopt the following definitions of concept and term:

Concept is a unit of thought.

Term is a lexical representation of a concept.

We suggest the use of "concept" instead of "object" and of "term" instead of " entity". We also prefer and agree to use "relationships" between concepts rather than "relations" between concepts.

Ontologies are used to describe and represent the domain knowledge. Ontology represents a shared understanding of domain knowledge and consists of concepts used to describe the domain knowledge and the relationships between those concepts. Formal axioms are used to constrain the interpretation and well-formed use of the defined concepts. Terms are used to designate the concepts and the relationships in the universe of discourse (Gruber 1993). Every branch of science has its own ontology.

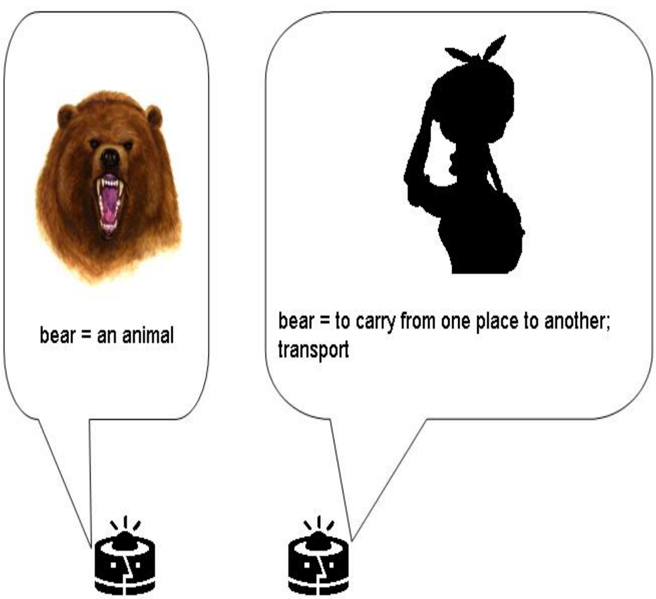


Fig. 3.2 Two different meanings of the same term- i.e. representing two different concepts

3.4 Ontology Commitments

In order to communicate effectively, users, applications and agents need to have a common and shared understanding of the terms used in communication. The terms need to have the same meaning for all parties involved in communication. For example, consider the word “bear”. For one agent, this term is associated with the concept of bear being an animal, while for the other agent this term is associated with the concept of the action of carrying something (Figure 3.2). How is it possible for these two different agents to reach an agreement if, for each of them, the same term has a different meaning? We need to ensure that all parties involved in the communication share the same ontology and have a common understanding of all terms used in conversation.

Ontological commitment is described as the agreement about the concepts and relationships between those concepts within the ontology (Gruber 1993). When a user, application or agent commit to an ontology, there is an agreement with respect to the meaning of the concepts and relationships represented. Furthermore, there is agreement to use the shared vocabulary coherently and consistently. Ontological commitment to the semantics (meaning) of the concepts and their relationships in the common ontology includes commitments to the axioms, rules, constraints etc. stated in the ontology.

Even though the Agent Communication Language (ACL) is generally used for agents’ communication (Labrou et al. 1999; Finin et al. 1994), different agents can communicate with each other using the same ontology. When different agents commit to an ontology, this common ontology defines the vocabulary and enables the different agents to exchange the information efficiently (Van Aart et al. 2002). The agents can meaningfully communicate about a domain even though they may be designed and used by different applications (Stuckenschmidt 2002).

It is also possible for an agent to commit to several ontologies. In some situations, an agent can commit to several ontologies at the same time where each ontology is used in communication with a particular agent. In other situations, different agents’ ontologies may contain knowledge complementary to each other. Those ontologies can then be merged into a large ontology that will serve as a common ontology for those agents. Of course, here it is very important to have these different ontologies formatted in the same way so that the agents can be designed in correspondence with this format and be able to easily accept and process the new ontologies.

Ontology commitments play an important role in the ontology design. We will give some examples to illustrate this. As we mentioned in the first chapter, the same term may have different meanings in different domains (Figure 1.5). Also, the same term can be defined differently within different ontologies from the same domain even though its meaning is the same; i.e. they refer to the same concept. For example, within Genetics the concept “gene” can be defined in different ways (Figure 3.3). For this reason, it is crucial that ontologies be designed by a number of domain experts together with ontology engineers. Ontology commitments need to be carefully considered and evaluated against the project objectives.

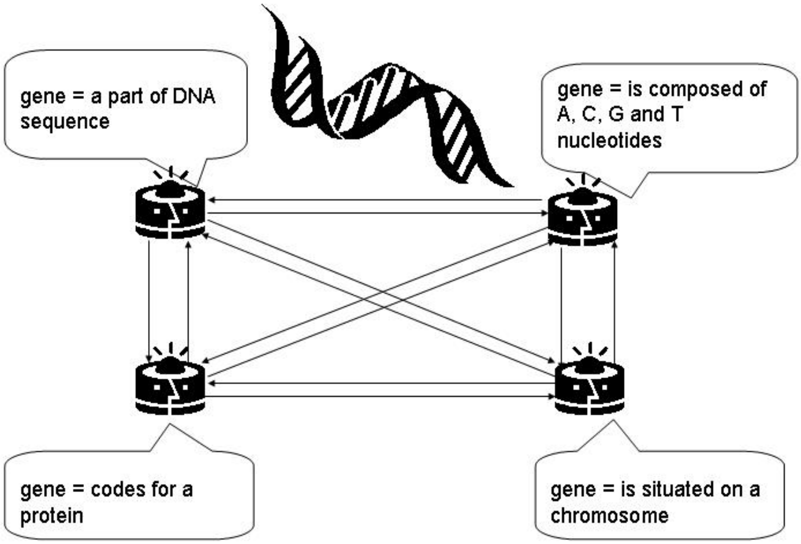


Fig. 3.3 Different definitions of the same concept

3.5 Ontology Community

Ontologies represent how a specific community understands part of the world. It is important for all members of the community to reach an agreement on (1) their understanding and definition of domain knowledge; and (2) representation of this knowledge through an ontology. A definition may be added to an ontology only with the consensus of all community members. No member can alter or override the definitions in an ontology according to his/her preferences. Community members, when committed to their common ontology, are able to use this ontology for various purposes such as the sharing and using of the available knowledge.

Different members of a community can share and reuse existing ontologies. Many ontologies are already available in electronic format. These ontologies can be imported into a new ontological development environment and can be reused by another application.

Any community can commit to ontologies developed by other communities. A group of communities can accept and commit to an ontology with more general terms and fewer constraints. These communities can use such general ontologies to develop their own specialized ontologies.

3.6 Generalization/Specialization of Ontologies

Generic ontologies contain less detailed information than do specialized ontologies. Usually, more detailed specialized ontologies are derived from the generic ontologies (Figure 3.4). These new specialized ontologies are more detailed as they refine and extend the general descriptions present in generic ontologies.

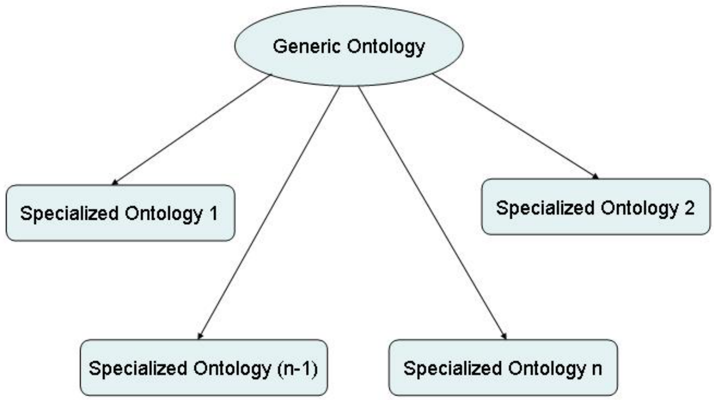


Fig. 3.4 Ontology specialization

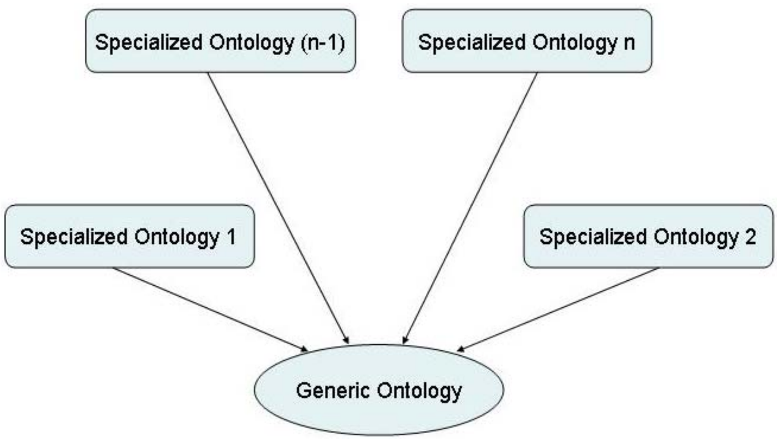


Fig. 3.5 Ontology generalization

Generic ontologies are more likely to be shareable by a large number of communities. Usually, specialized ontologies are for a smaller community within a larger community. A generic ontology consists of a minimal number of axioms, while a specialized ontology has a large number of axioms and needs a very expressive language. It is possible to progress gradually from generic to specialized ontologies through an incremental increase of the number of axioms.

Adding constraints can specialize an ontology to be used by a specific community. In the "Ontolingua", a library of ontologies, there are ontologies with general terms and minimum constraints that can be used by many communities to develop specialized ontologies (Farquhar et al. 1997). One can add new terms along with their definitions to specialize an ontology for a subcommunity (Visser et al. 1998).

We can also build a generic ontology which is based on a number of specialized ontologies. Such a process is referred to as "ontology generalization" and is shown in Figure 3.5. This process aims to resolve the conflicts between definitions of terms in the ontologies. The result of such a process is an ontology that can be accepted by the participating communities.

3.7 Properties of Ontologies

Some essential properties of ontologies are:

- ontologies describe a specific domain
- ontology users agree to use the ontology terms consistently
- ontology concepts and relations are unambiguously defined in a formal language by axioms and definitions
- relationships between ontology concepts determine the ontology structure e.g. hierarchical or non-hierarchical. The generalization/specialization relationship (i.e. "is-a" relationship) between two concepts is an example of a hierarchical relationship between concepts.
- ontologies can be understood by computers

Ontology has different meanings within different contexts (Figure 3.6). In Philosophy and Metaphysics, the ontology encompasses nature and existence, beings and relations between beings. The resulting knowledge is explicit, shared and easily understandable by humans. In Computer Science and Artificial Intelligence, an ontology is used to formally and explicitly represent shared domain knowledge through definitions and axioms of concepts and relationships between the concepts. The main difference is that in Computer Science and Artificial Intelligence literature, ontology is designed to be understandable by machines.

Two important functions of ontologies are that they:

1) enable agents to work cooperatively to communicate with each other
2) make the available information more accessible to automated agents

Agents working cooperatively can communicate with each other by means of a common ontology. Ontology is machine readable, and supports agent communication by defining and providing a shared vocabulary to be used in the course of communication. Ontologies not only provide a definition of the terms that can be used in communication; ontologies also provide the definition of the world in which an agent grounds its actions. Different agents of a system can reach a shared understanding by committing to the same ontology. This enables them to make statements, communicate knowledge and make different queries. Use of ontology permits coherent communication and easier information sharing between different agents, enabling agents to cooperate and coordinate their actions.

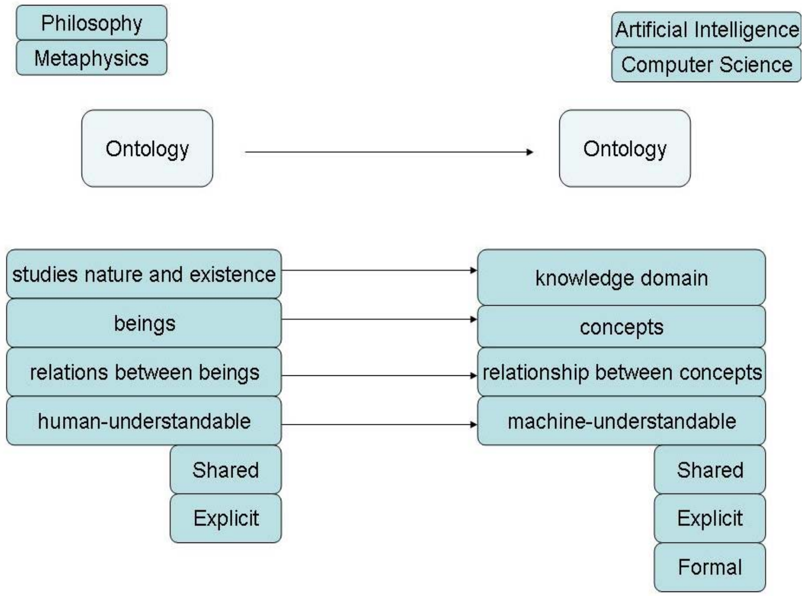


Fig. 3.6 Meaning of ontology in two different contexts

If the available information is to be more accessible to automated agents, it is essential that the meaning of this information be understood by such agents. People and software systems would be able to extract, analyze and apply information from different organizational sites if those sites operate on the basis of the same underlying ontology. Adding information that describes Web content in a machine-understandable way enables Web information resources to become more accessible to automated processes (Horrocks et al. 2002). Typically, an ontology has a hierarchical structure, and describes domain concepts and their properties. As such, an ontology is a collection of shared and precisely defined terms that can be used in meta-data to describe Web content in a machine-understandable manner. Through this, ontologies play an important role in efficient and effective information access and retrieval.

3.8 Characteristics of Ontology Models

Analogous to the characteristics of the data models (Chandrasekaran et al. 1999), we developed a similar scheme for the ontology models:

- Concept is a unit of thought. Concepts may have abstract meaning (e.g. human diseases) and physical meaning (e.g. gene).
- Term is a lexical representation of a concept (or name of a concept).

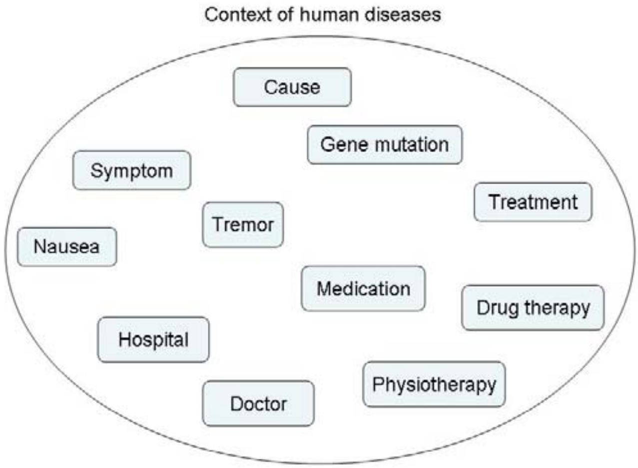


Fig. 3.7 Several concepts in the context of human diseases

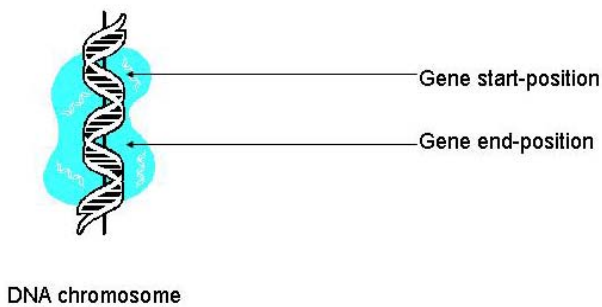


Fig. 3.8 Gene is situated on a DNA chromosome

- Context is used to group terms in an ontology. For example, in the context of human diseases, we group terms used to describe and represent the knowledge of human diseases (Figure 3.7).
- Terms exist in various relationships with each other. For instance, the term ‘gene’ is related to the term ‘DNA chromosome’ because a gene is situated on a DNA chromosome (Figure 3.8).
- The knowledge representation changes over time. The level of detail of the ontologies is related to the level of detail of the information. It is often necessary to filter and omit details of information in order to make the situation clearer by representing only information relevant to the concept within the context. At other times, this is inadvisable because the requested knowledge can be found only in the detailed information (Langacker 1987). For example, when determining the exact position of a gene within human DNA, we firstly

locate a chromosome (out of 23 different human chromosomes) that contains this gene. At this stage, the ontology associates only a chromosome with the term "gene position". Secondly, we locate an arm of this chromosome (p or q arm) where the gene of interest is situated. At this stage, the ontology associates p or q chromosome arm with the term "gene position". Finally, we locate the exact region within the chromosome arm, such as p11. At this final stage, the ontology associates the exact region of a chromosome with the term "gene position". The knowledge model that describes the position of a gene within human DNA needs to be updated as new knowledge becomes available. Generally, the knowledge model needs to be updated when new knowledge emerges.

- In processes such as information retrieval, information needs to be integrated at different levels of detail. It may be necessary to navigate at different levels in the ontology structure, select the relevant information and integrate this information in a controlled way. Information may need to be combined at different levels of detail but merged on a specific level. A mechanism for integrating ontologies should provide these services.
- Incorporation of a new concept within an ontology may require the presence of an additional concept. Similarly, the deletion of an ontology concept may make the presence of another ontology concept redundant. For example, gene mutation is a change of the gene structure which may cause a disease. If we introduce the term 'gene mutation' into an ontology, we may also need to introduce the term 'human disease'. Because the ontological model is a semantic network of concepts and their relationships, the model extension may change the ontology structure significantly.

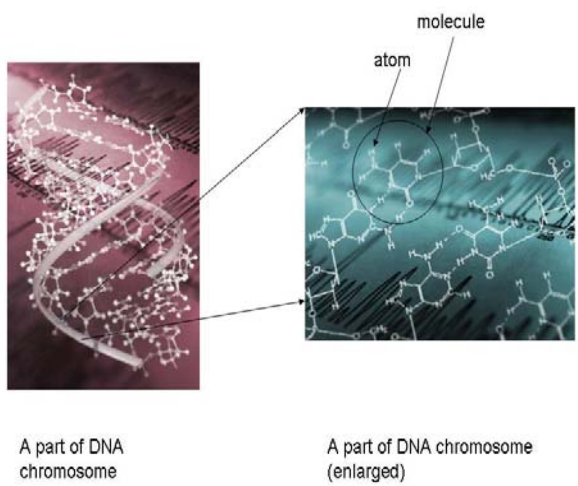


Fig. 3.9 Double-helix DNA is composed of molecules which are composed of atoms

- Different terms may be incorporated into the ontology to enable the definition of a new term. For example, as we show in Figure 3.9, human DNA is a double-stranded helix composed of four different nucleotides that consist of a base (Adenine (A), Cytosine (C), Guanine (G) and Thymine (T)), a phosphate molecule and a sugar molecule (deoxyribose). The nucleotides are grouped together in a helix forming human DNA. Different atoms are grouped together to form a molecule and different molecules are grouped together to form DNA. The level of detail described by an ontology will depend on the purpose of the ontology design.

3.9 Representation of Ontology Domain

Hierarchies have been predominantly used to represent ontologies. Two main reasons for this are: (1) hierarchies are similar to the way we organize the mental models of the world in our minds; and (2) hierarchies allow for the generalization/specialization mechanism in information processing.

In the example given in Figure 3.10, we represent the top-level of the 'Nutrients' ontology that describes the basic classification of nutrients as carbohydrates, proteins and fats. At the top of the hierarchy, we have the concept "nutrients". On the next level down, we have three concepts: "carbohydrates", "proteins" and "fats". The reason for the choice of these particular three different groups to classify the concept "nutrients" depends on many factors such as the purpose and use of the ontology. We will discuss this in Chapter 7. The ontology has further branches as follows: "carbohydrates" can be "starch", "sugar" or "fibre", "proteins" can be "incomplete protein" or "complete protein"of vegetable or animal protein respectively. "Fats" can be "saturated fats", "unsaturated fats", "dietary cholesterol" and "trans fatty acids". The knowledge in the 'Nutrients' ontology is represented through hierarchical relationships between different concepts.

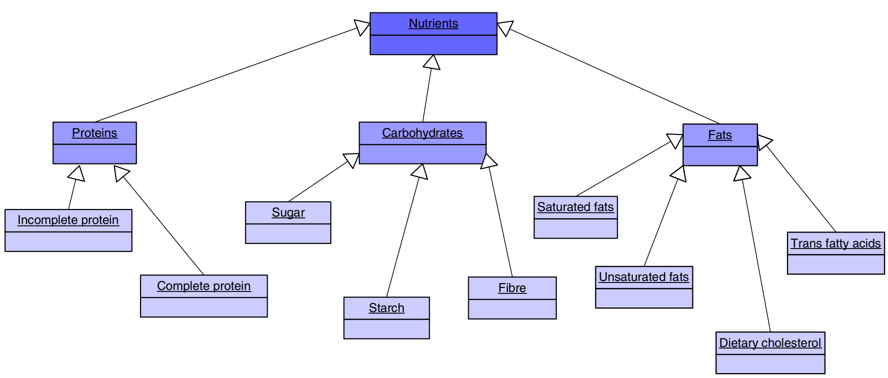


Fig. 3.10 Top-level of 'Nutrients' ontology

Although ontologies often assume the form of a taxonomic class hierarchy, they are not restricted to hierarchies in any way. In fact, ontologies may take on much more general and complex structures (Gruber 1995). The ontology concepts may be grouped together to form layers of related hierarchy trees, networks, cubical structures or any other shapes. The way the ontology concepts are grouped together depends on the purpose of the ontology design and on the complexity of the knowledge that it represents.

3.10 Ontology Design Versus Data Modelling

Ontologies and data models both represent domain knowledge but exhibit some fundamental differences. Useful points of reference when making the comparison are (Spyns et al. 2002):

1) Application dependencies
2) Knowledge coverage
3) Expressive power
4) Operation levels

Ontology and data models differ from each other in their dependencies on the application(s) they are designed for. Data models are designed to fulfil specific needs and therefore depend on the specific tasks that need to be performed. Users, goals, purposes and intended use of the model influence the modelling process and the level of detail described by this model. In contrast, ontologies are as generic and task-independent as possible. Minimum of application requirements are being considered during the ontology design. For example, an important property of a gene is the fact that gene codes for a protein. If the application does not require this knowledge, the data model will be designed without it. On the other hand, ontologies represent agreed and shared knowledge. So, the fact that gene codes for a protein will need to be included in the gene ontology. Ontology consists of relatively generic knowledge that can be reused by different kinds of applications across a community of users.

An ontology and data models differ from each other in the amount and kind of knowledge they cover. Data models focus on the establishing correspondence between organization of the data in databases and concepts for which the data is being stored. On the contrary, ontologies are concerned with the understanding of the knowledge by community members. Ontologies are a more complete representation of concepts and relationships. Ontology knowledge is considered to be the hard coded part of a computer system (Greiner et al. 2001). Knowledge understood to be true within a certain domain does not change relatively to the way this knowledge is organized and stored within a computer system. The data represent a dynamic part of a computer system which may change even during run-time (Greiner et al. 2001).

Ontologies have more expressive power compared to the data models. Data modelling languages use typical language constructs. In contrast, ontology languages are more expressive languages as they include constructs that express other kinds of meaningful constraints such as taxonomy or inferencing. Ontology languages make the domain conceptualization more correct and precise.

Ontology and data models operate on different levels. Ontologies are generic, task- and implementation-independent and as such operate on a higher level of abstraction. In contrast, data models are at the lower level of abstraction. Thus a ‘new car database’ would model a car using attributes such as model and maker of car, pricing, colour, engine size, accessories. In contrast, a ‘repair database’ might model spare parts associated with a particular model and make of a car, their price, the price for the labour to fit in the spare part, result ranges for certain tests, etc. Note each of these data modelling have a narrow view of the concepts of a car. As the ontology is the knowledge shared by different applications across a community, we can design a Car Ontology which will represent the knowledge common and shared by both ‘new car database’ and ‘repair database’. This knowledge will operate on a higher level of abstraction and will be designed independently from the applications running on each of these databases.

Differences between the ontologies and data models, the characteristics of application dependencies, knowledge coverage, expressive power and operation levels, can be used by data modellers and ontology designers as points of reference. These guidelines will help them stay focused on their own goal.

Because ontologies and data model operate on the different levels but are oriented in the same direction, it is possible to:

1) use ontologies to support data model design
2) use data models to elicit knowledge that could be useful for ontology design
3) integrate ontologies with data models
4) use ontologies as means of working with and querying databases with different data models in the same domain (see Figure 1)

Ontologies can be used to build, support and clarify data models. An ontology can play an important role during the data model design, especially in the requirements analysis. A wide range of ontologies is available via ontological libraries and can be used to assist in the modelling of a specific application domain.

Also, data models may support the ontology design process. Data models may help to provide knowledge that will help designers make decision regarding organization and structuring of the concepts within an ontology. The ontologies designed through support of data models are usually specialized ontologies and not general ontologies. The number of applications for which these specialized ontologies can be used is limited.

The integration of ontologies with data models into a unified meta-model can result in an effective combination of data, documents and formal knowledge (Kuhn and Abecker 1997). The main goal of this integration is to enrich the data models and increase the expressiveness of the domain under analysis. The application designer needs to identify ontologies from the ontological libraries that are suitable to be used for a portion of the domain which needs to be formally described.

3.11 Ontology Versus Knowledge Base

The difference between an ontology and a knowledge base can be seen in their different objectives (Maedche 2003). An ontology captures and represents the conceptual structure of a domain in a form that is shared by the community of users in that domain. In contrast, a knowledge base (Dillon and Tan 1993) seeks to model knowledge in that domain in a form suitable for a particular Knowledge-Based System (KBS) to carry out problem solving. Knowledge base models the problem solver's (expert's) approach to problem solving for the particular set of problems being addressed within this domain. An ontology has terminology and conceptualization agreed by the community to be correct and to be consistently used across the community. Whilst in the KBS, the terminology can be specific to the particular system and consistency is understood to mean logical consistency within the narrow focus of the problems being solved. The knowledge base enables an inference engine or agent to reason in order to solve the particular set of problems that the KBS is meant to address.

The main differences between ontology and knowledge bases are summarized in Table 3.1.

Table 3.1 Ontology versus knowledge base

	Ontology	**Knowledge base**
objectives	conceptual structures of a domain	particular states of a domain
consistency	facts always true	facts true for a particular state of affairs
actions	communication	problem solving
knowledge	domain knowledge	operational knowledge & restricted domain knowledge
applicability	shared by community	specific to system

The purpose of an ontology is to describe facts assumed to be always true by the community of users. It aims to capture the conceptual structures of a domain. In contrast, a knowledge base aims to specify the portion of the domain that is useful for solving a particular set of problems. It may also describe facts related to a particular state of affairs.

For an agent, a shared ontology describes a vocabulary for communicating about a domain. In contrast, a knowledge base contains the knowledge needed to solve problems or answer queries about such a domain. Domain knowledge is described by ontologies while operational knowledge is described in the knowledge base. An ontology specifies knowledge about a certain domain while a knowledge base specifies knowledge used by the agent and/or an inference engine to perform its actions.

Recently, the use of ontologies in preference to knowledge bases for the agents' operational knowledge has been proposed. This would result in two different kinds of ontologies: domain ontology and operational ontology (which replaces the knowledge base). Let us consider an example to clarify the difference between domain and operational knowledge. We may be assigned the task of delivering some books to a certain address. We may have the domain knowledge about the

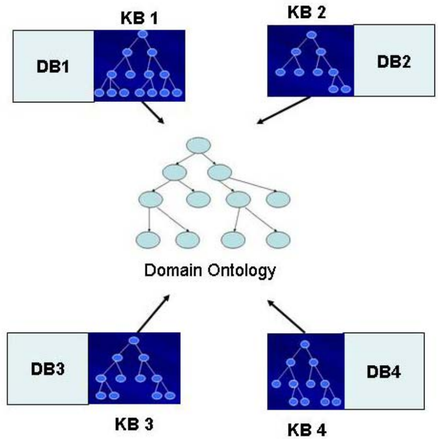


Fig. 3.11 Different databases share same ontology

street locations in the city. Still, we may not able to complete this task if we lack the operational knowledge (i.e. if we do not know how to perform the action of locating the address and delivering the books). Domain and operational knowledge are both needed by an agent if it is to function optimally.

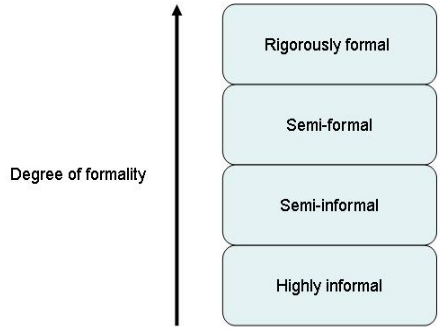
In Figure 3.11, we show an example where each database (DB) of the four different databases has specific applications associated with it and uses its own knowledge base (KB) to perform the specific actions. The knowledge base of one database is independent from the knowledge base of another database. A new application is proposed where the four different databases need to collaborate with each other. A domain ontology needs to be designed for this purpose and each of the four databases needs to commit to this common ontology. This ontology would enable effective and efficient cooperation and collaboration of the four databases. The databases would be able to communicate with each other and share the information, tasks and results efficiently.

3.12 Classification of Ontologies

Different types of interpretation are associated with the concept of ontology. The knowledge expressed by different ontologies might be the same, but the ontologies may be different from each other in the way in which this knowledge is expressed. The ontologies may differ from each other in regard to their:

1) degree of formality
2) degree of granularity
3) level of generality
4) amount, type and subject of conceptualization
5) expressiveness

Fig. 3.12 Classification of ontologies according to degree of formality



3.12.1 Degree of Formality

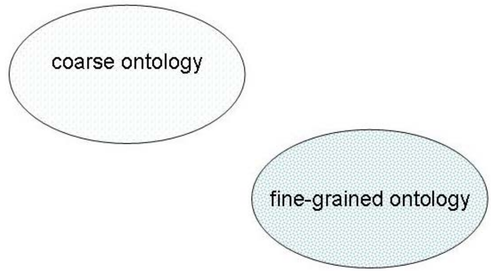
Ontologies differ in the degree of formality with which the terms and their meaning are expressed (Uschold et al. 1996). As shown in Figure 3.12, ontologies are classified into following four categories:

1) highly informal ontologies
2) semi-informal ontologies
3) semi-formal ontologies
4) rigorously formal ontologies

- Ontologies expressed in natural language are "highly informal" ontologies. Due to the ambiguity of natural language, term definitions in "highly informal" ontologies may be ambiguous, and therefore difficult for a software agent to work with.
- Ontologies expressed in a restricted and structured form of natural language are "semi-informal" ontologies. Restricting and structuring of natural language helps to reduce the ambiguity and improve the clarity.
- Ontologies expressed in artificial languages which are formally defined are "semi-formal" ontologies.
- Ontologies whose terms are precisely defined through formal semantics and theorems, and have desired ontology properties (such as soundness and completeness) are "rigorously formal" ontologies.

Having an informal ontology makes it easier to understand the motivation and idea behind the ontology design, and to discuss possible modifications that need to be introduced into the ontology. The computers cannot use this informal ontology. In order for an ontology to be used by computers and by various applications, it needs to reach a stage where it is formally described. Only when formally described, can an ontology reach its ultimate goal of being used by some program/application(s).

Fig. 3.13 Classification of ontologies according to degree of granularity



3.12.2 Degree of Granularity

The ontologies can be classified according to their granularity (see Figure 3.13) in the representation of domain concepts into:

1) coarse ontologies
2) fine-grained ontologies

A coarse ontology is shared among users who already agree on the underlying conceptualization of the domain. It consists of a minimal set of axioms written in a language of minimal expressivity and supports only a limited set of specific services.

A fine-grained ontology is used to establish agreement on the underlying conceptualization of the domain between the ontology users. In contrast, fine-grained ontology consists of a large number of axioms written in a language of extensive expressivity, to support a variety of services.

3.12.3 Level of Generality

The ontology types, shown in Figure 3.14, are classified into four groups according to the level of generality used in the description of a domain (Guarino 1998):

1) application ontologies
2) task ontologies
3) domain ontologies
4) top-level ontologies

- Vocabulary related to both a particular domain and a particular task is described by “application” ontologies. They are often a specialization of both domain and task ontologies.
- Vocabulary related to a specific task or activity is described by “task” ontologies.
- Vocabulary related to a particular domain is described by “domain” ontologies.
- General concepts or common-sense knowledge is described by “top-level” ontologies. Those ontologies are independent of a particular problem or domain.

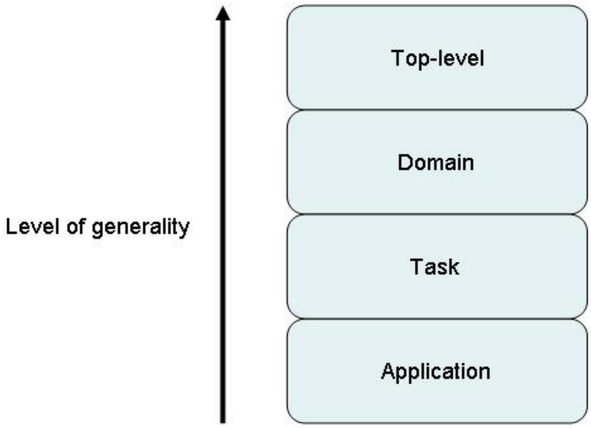


Fig. 3.14 Classification of ontologies according to level of generality

This kind of classification could confuse someone who is new to the ontology field. By the ontology definition, an ontology is used to represent the knowledge of a particular domain and all ontologies are "domain" ontologies as they describe the domain knowledge. It could also be confusing to say that the "top-level" ontologies are independent of a particular domain as this contradicts the ontology definition. Also, an ontology is designed for the purpose of ultimately being used by an application. In order to avoid any confusion associated with this ontology classification, we need to remain aware of the context within which this ontology classification occurs.

3.12.4 Amount, Type and Subject of Conceptualization

Ontologies can be classified according to amount and type of conceptualization structure and subject of the conceptualization (Van Heijst et al. 1997).

1. The amount and type of structure of the conceptualization is mainly concerned with the level of granularity of the conceptualization. In Figure 3.15, we show ontology types associated with this classification:

1) terminological ontologies
2) information ontologies
3) knowledge modelling ontologies

- "Terminological" ontologies are not really ontologies but lexicons used to specify terminology for representation of the domain knowledge. But terminological ontologies do not capture semantics of the terms.
- "Information" ontologies specify the record structure of databases and enable their controlled use and management. But information ontologies do not define the concepts that are instantiated by database instances.

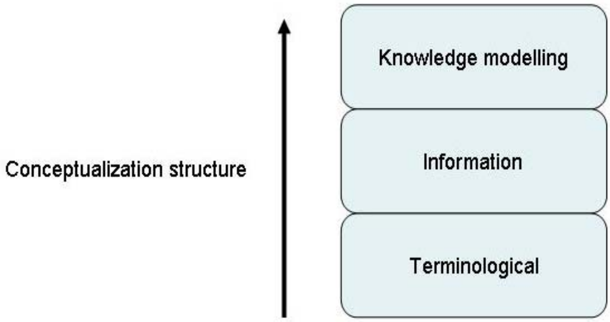


Fig. 3.15 Classification of ontologies according to conceptualization structure

- "Knowledge modelling" ontologies specify conceptualizations of knowledge. They are often specified according to a particular use of the knowledge they describe. Knowledge modelling ontologies are structurally richer than information ontologies.

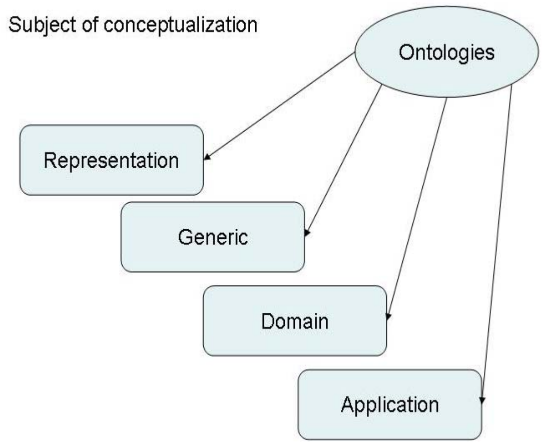
In the early stages of ontology acceptance by the computer and information science community, the definition of an ontology and the purpose of its design should be clearly stated. Why do we compromise the ontology definition? If something is really not an ontology (like "terminological ontologies" mentioned above), why do we support the existence of these kinds of "ontologies" if they are not ontologies? In doing so, we only dilute and hide the real importance of an ontology. The concept of ontology needs to stand independently from other similar concepts (such as lexicons) in order for its true and full significance to be appreciated. If we hold back on what we already know, we cannot fully progress towards the point where the ontology is taking us. Moreover, why do we need to define a class of "information ontologies" if the same can be done by UML or ORM modelling techniques? By using ontology for purposes other than those for which it was introduced, the full significance of the ontology technology is obscured and prevented from achieving its full potential. We need to embrace the true ontology definition, know the ontology's purpose and reason for its existence.

2. The subject of the conceptualization concerns the type of knowledge that is modelled in the ontologies, and is further subdivided into:

1) application ontologies
2) representation ontologies
3) domain ontologies
4) generic ontologies

- "Application" ontologies contain concepts necessary for knowledge modelling required for specific applications. Usually, these ontologies take terms from more general ontologies and extend this knowledge by representing method- and task-specific components.

Fig. 3.16 Classification of ontologies according to subject of classification



- "Domain" ontologies contain concepts specific to a particular domain. Domain ontologies specify constraints to be applied on the structure and the content of the domain knowledge. The design of domain ontologies is based, and depends on, the domain knowledge.
- "Generic" ontologies specify concepts that are generic across many fields. Specialization of "generic" ontologies for a particular domain results in " domain" ontologies.
- "Representation" ontologies make no claims about the world and are neutral with respect to the world. They provide a representational framework; " domain" and "generic" ontologies can be described using the primitives given in the representation ontologies.

This classification is shown in Figure 3.16. This classification according to the subject of the conceptualization is similar to the classification according to the level of generality which is shown in Figure 3.14.

3.12.5 Expressiveness of Ontologies

Ontologies can be classified on the grounds of their expressiveness (Lassila and McGuinness 2001) into the following categories:

1) controlled vocabularies
2) glossaries
3) thesauri
4) informal is-a hierarchies
5) formal is-a hierarchies
6) formal instances relations ontologies
7) frames ontologies
8) value restriction ontologies
9) general logical constraints ontologies

- "Controlled vocabularies" are the simplest possible notion of ontology. They are nothing more than a finite list of terms with an unambiguous interpretation.
- "Glossaries" are also lists of terms, but their meanings are ambiguously defined. "Glossaries" are usually expressed in natural language statements; they are aimed at humans and cannot be used by computer agents.
- "Thesauri" define the relationships between terms and, in this way, add semantics to glossaries. "Thesauri" do not provide an explicit hierarchical structure, but broader or narrower term specifications can be used to deduce this kind of structure. A computer agent can often interpret the relationships defined in a thesaurus.
- "Informal Is-a hierarchies" provide a general notion of generalization and specialization. But the inheritance cannot be assumed here because concepts of "Informal Is-a hierarchies" are not organized according to "strict subclass hierarchies". This means that an instance of a more specific class is not necessarily an instance of a more general class.
- "Formal Is-a hierarchies" are ontologies where concepts form "strict subclass hierarchies" and the inheritance is always applicable; namely, an instance of a more specific class is also an instance of a more general class.
- "Formal instances relations" ontologies enforce also a strict hierarchical structure. Formal instance relations hold when the ontology content describes ground individuals and their relationships with the concepts they instantiate.

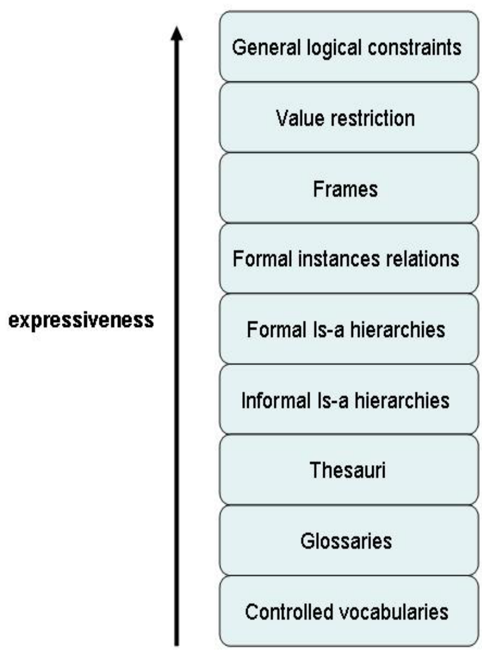


Fig. 3.17 Classification of ontologies on the grounds of their expressiveness

- "Frames" ontologies describe concepts together with their properties. Inheritance can be applied to these properties; more specific concepts down the hierarchy can inherit properties specified for a more general concept.
- "Value restriction" ontologies apply restrictions on the values associated with properties. These restrictions are usually inherited by more specific concepts down the hierarchy.
- "General logical constraints" ontologies are written in very expressive ontology languages permitting a thorough specification of concepts and their properties. Of all the abovementioned ontology types, "general logical constraints" ontologies have the richest expressiveness.

The classification of ontologies according to their expressiveness is shown in Figure 3.17. In this book, classes that cannot be understood by computers such as "controlled vocabularies", "glossaries", some "thesauri" will not be classified as ontologies.

3.13 Conclusion

In this chapter, we focused on the following:

- ontology origins; we compared the meaning of an ontology in Philosophy and Computer Science, and showed main similarities and differences;
- ontology definition; we emphasized the importance of (1) formal representation of the knowledge in order to make the ontology understandable by computers, and (2) agreement in order to make the ontology sharable within the ontology community;
- ontology commitments are another important factors in ontology sharing; users, agents and applications need to commit to the common ontology and agree to share this ontology in a coherent manner;
- generalization versus specialization of ontologies; specialized ontologies contain more details then the generalized ontologies. As a consequence, only a limited number of users can commit to specialized ontologies while the generalized ontologies are designed for a broader community;
- main ontology properties and characteristics of ontology models in terms of its basic elements (e.g. concepts, terms, their relationships, ontology context) as well as more complex ontology features (e.g. knowledge representation, level of detail and ontology extension);
- representation of an ontology domain through hierarchies and other structures;
- distinction of (1) ontology from knowledge base and (2) ontology model from data model; we summarized the main points of reference when doing the comparisons;
- four different types of ontology classification according to the degree of formality, degree of granularity, generality level, expressiveness, and amount, type and subject of conceptualization; we discussed the relevance of this classification in regard to the ontology definition.

In the following chapter, we will discuss some major agent design methodologies to cover different aspects of the design process.

References

1. Greiner, R., Darken, C., Santoso, N.I.: Efficient reasoning. ACM Computing Surveys 33, 1–30 (2001)
2. Chandrasekaran, B., Josephson, J., Benjamins, V.: What Are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems 14, 20–26 (1999)
3. Corazzon, R.: Ontology. A resource guide for philosophers (2000), `http://www.formalontology.it` (retrieved: May 20, 2003)
4. Dillon, T., Tan, P.: Object-Oriented Conceptual Modeling. Prentice Hall, Australia (1993)
5. Farquhar, A., Fikes, R., Rice, J.: The Ontolingua Server: A tool for collaborative ontology construction. International Journal of Human–Computer Studies 46, 707–727 (1997)
6. Fensel, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Springer, Heidelberg (2001)
7. Finin, T., Fritzson, R., McKay, D., McEntire, R.: KQML as an agent communication language. In: The third international Conference on Information and Knowledge Management (CIKM 1994), pp. 456–463 (1994)
8. Gómez-Pérez, A.: Knowledge Sharing and Reuse. The Handbook on Applied Expert Systems (1998)
9. Gruber, T.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human and Computer Studies 43, 907–928 (1995)
10. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5, 199–220 (1993)
11. Guarino, N.: Formal Ontology in Information Systems. In: The conference on Formal Ontology in Information Systems (FOIS 1998), pp. 3–15 (1998)
12. Horrocks, I., Patel-Schneider, P.F., Van Harmelen, F.: Reviewing the Design of DAML+OIL: an Ontology Language for the Semantic Web. In: The Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI 2002), pp. 792–797 (2002)
13. Kuhn, O., Abecker, A.: Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges. Journal of Universal Computer Science 3, 929–954 (1997)
14. Labrou, Y., Finin, T., Peng, Y.: Agent Communication Languages: The Current Landscape. IEEE Intelligent Systems 14, 45–52 (1999)
15. Langacker, R.W.: Foundations of Cognitive Grammar: Theoretical Prerequisites. Stanford University Press (1987)
16. Lassila, O., McGuinness, D.: The role of frame-based representation on the semantic Web: area The Semantic Web. Electronic Transactions on Artificial Intelligence (ETAI) Journal 6(5) (2001)
17. Maedche, A.D.: Ontology Learning for the Semantic Web. Kluwer Academic Publishers, Dordrecht (2003)
18. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice-Hall, Englewood Cliffs (1995)
19. Spyns, P., Meersman, R., Jarrar, M.: Data modelling versus Ontology engineering. SIGMOD Record 31, 7–12 (2002)

20. Stuckenschmidt, H.: Exploiting Partially Shared Ontologies for Multi-Agent Communication. In: Klusch, M., Ossowski, S., Shehory, O. (eds.) CIA 2002. LNCS, vol. 2446, pp. 249–263. Springer, Heidelberg (2002)
21. Uschold, M., Gruninger, M.: Ontologies: principles, methods, and applications. Knowledge Engineering Review 2(11), 93–155 (1996)
22. Van Aart, C., Pels, R., Caire, G., Bergenti, F.: Creating and Using Ontologies in Agent Communication. In: The first International Joint Conference on Autonomous Agents and Multi-agent systems (AAMAS 2003) (2002)
23. Van Heijst, G., Schreiber, A.T., Wielinga, B.J.: Using explicit ontologies in kbs development. International Journal of Human-Computer Studies 46, 183–292 (1997)
24. Visser, P.R.S., Jones, D.M., Bench-Capon, T.J.M., Shave, M.J.R.: Assessing heterogeneity by classifying ontology mismatches. In: Formal Ontology in Information Systems (FOIS 1998), pp. 148–162 (1998)