

Chapter 2

An Introduction to Ontologies and Ontology Engineering

Catherine Roussey, Francois Pinet, Myoung Ah Kang, and Oscar Corcho

2.1 Introduction

In the last decades, the use of ontologies in information systems has become more and more popular in various fields, such as web technologies, database integration, multi agent systems, natural language processing, etc. Artificial intelligent researchers have initially borrowed the word “ontology” from Philosophy, then the word spread in many scientific domain and ontologies are now used in several developments. The main goal of this chapter is to answer generic questions about ontologies, such as: Which are the different kinds of ontologies? What is the purpose of the use of ontologies in an application? Which methods can I use to build an ontology?

There are several types of ontologies. The word “ontology” can designate different computer science objects depending on the context. For example, an ontology can be:

- a thesaurus in the field of information retrieval or
- a model represented in OWL in the field of linked-data or
- a XML schema in the context of databases
- etc.

C. Roussey (✉)

LIRIS, Université de Lyon, France

e-mail: Catherine.Roussey@cemagref.fr

F. Pinet

French Institute for Agricultural and Environmental Engineering, Clermont Ferrand, France

e-mail: Francois.Pinet@cemagref.fr

M.A. Kang

Laboratory of Computer Science, Modelling and System Optimization, Blaise Pascal University, Clermont Ferrand, France

O. Corcho

Departamento de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid (UPM), Spain

e-mail: ocorcho@fi.upm.es

It is important to distinguish these different forms of ontologies to clarify their content, their use and their goal. It is also needed to define precisely the vocabulary derived from the word ontology. For example what is the difference between a core ontology and a domain ontology? First, we introduce and define the different types of ontologies. Second, we present some methodologies to build ontologies. Some of the illustrative examples will be taken from project presentations made in the context of the COST UCE Action C21 (Urban Ontologies for an improved communication in UCE projects TOWNTOLGY) or, in general, in the area of Geographic Information Systems (GIS).

2.2 Ontology Classifications

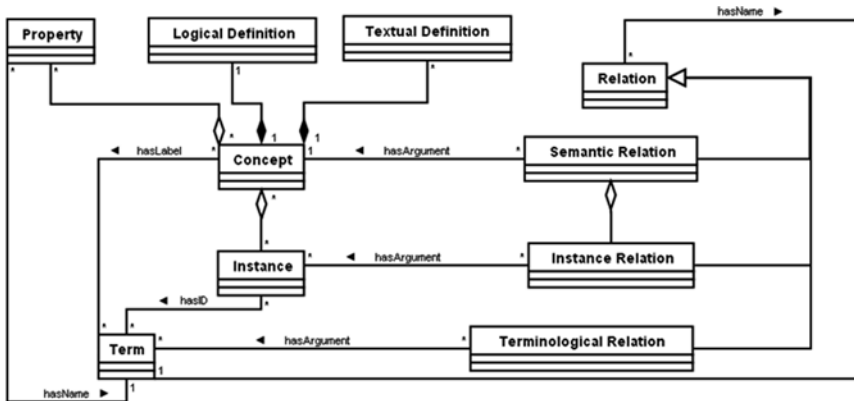
Several classifications of ontologies have been presented in the literature (Lassila and McGuinness 2001; Gomez-Perez et al. 2004; Borgo 2007, etc). Each of them focused on different dimensions in which ontologies can be classified. This section focuses on two of these classifications: the first one classifies ontologies according to the expressivity and formality of the languages used: natural language, formal language, etc.; the second one is based on the scope of the objects described by the ontology.

2.2.1 Classification Based on Language Expressivity and Formality

Depending on the expressivity of an ontology (or, in general, of a knowledge representation language), different kinds of ontology components can be defined (concepts, properties, instances, axioms, etc.). Figure 2.1 presents the set of components that we will use to provide our classification based on language expressivity. For example, if we focus on concepts, which are one of the main components of ontologies, the UML class diagram of Fig. 2.1 shows as that they can be defined in different (and complementary) ways:

- By their textual definitions: For example the concept “*person*” is defined by the sentence “*an individual human being*”,
- By a set of properties: for example the concept “*person*” has the property “*name*”, “*birth date*” and “*address*”; note that a property can be reused for several concepts.
- By a logical definition composed of several formulae: for example the concept “*person*” is defined by the formula “ $LivingEntity \cap MovingEntity$ ”.

A concept can also be defined by the set of instances that belong to it. For example, “*Martin Luther King*” is an instance of the concept “*person*”. This last definition is called the extensional definition of a concept and the three former definitions are called intensional definitions of a concept.



Legend:

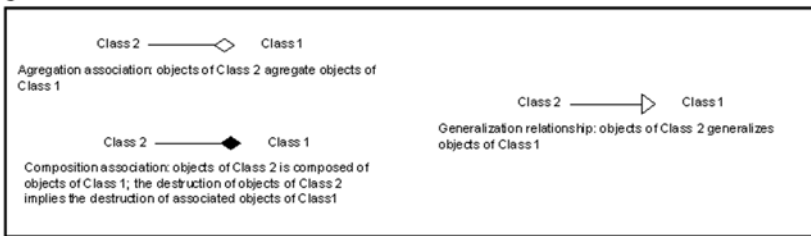


Fig. 2.1 UML class diagram representing ontology components and their relationships

Concepts, instances and properties are referenced by one or more symbols. Symbols are terms that humans can rapidly understand roughly by reading them. And finally all these ontology components are connected through relations. Semantic relations link only concepts together: for example the location relationship indicates that city concept is localized in a country concept. Instance relations connect only instances and instance relations are often instances of semantic relations, although it is not always the case. Some relations between instances can be contextual and cannot be generalized to all instances of their concept. An example of instance relation is that the city instance named Paris is localized in the country instance named France. All cities are localized in a country. A contextual instance relation can be that the person instance named “John Travolta” is localized in the city instance named “Paris” at the point in time 31 January 2010. The terminological relations express the relationships that terms can have: for example the term “*person*” is synonym to the term “*human being*”

According to the usage of these components, in the following sections we present four kinds of ontologies. In each section we explain which type of language is normally used to define the ontology and we provide some examples for illustration purposes. The classification starts using the less formal languages to the more formal one.

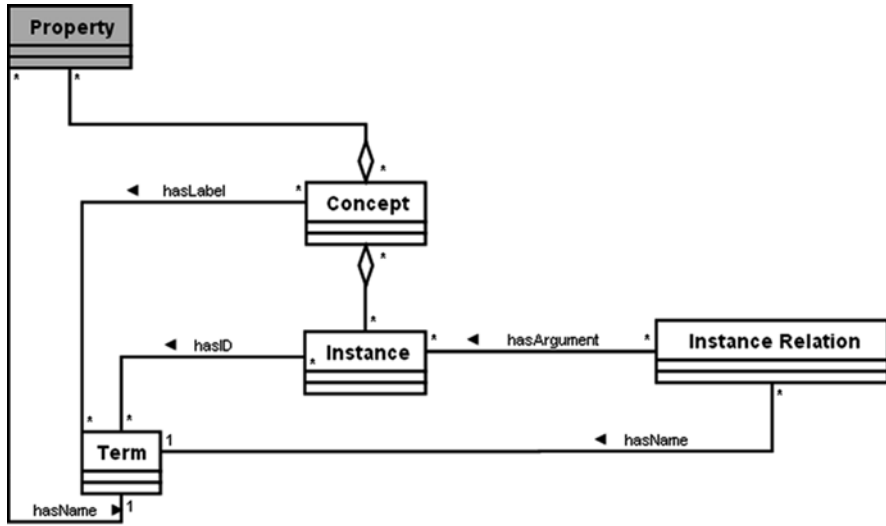


Fig. 2.2 UML schema of information ontology component and their relationships

2.2.1.1 Information Ontologies

Information ontologies are composed of diagrams and sketches used to clarify and organize the ideas of collaborators in the development of a project. These ontologies are only used by humans. The characteristics of information ontologies are:

- Easily modifiable and scalable
- Synthetic and schematic
- They are normally used during a design process of a project: for example, information ontology can be used during the conception phase of information system development project or during the design of floor plan in architectural construction project.

As shown in Fig. 2.2, information ontologies focus on concepts, instances and their relationships. Their goal is to propose an overview of a current project in order to express the state of this project. The grey color of the property elements means that properties are not always well defined by information ontologies.

Information ontologies are normally described by means of visual languages, so that they can be easily understood by humans. A Mind Map is a good example of this type of visual language. For example the OnToKnowledge project about methodology for ontology design propose to add a Mind Map plug-in called Mind2Onto in their ontology editor called OntoEdit (Sure and Studer 1999). They notice that Brain Storming is a good method to quickly and intuitively start a project. Their Mind Map plug-in is a support for discussion about ontology structure. Mind Map descriptions will be followed by three examples of information ontologies: one example will be taken from urban planning project, another one come from architectural design and the latter is used in a construction project.

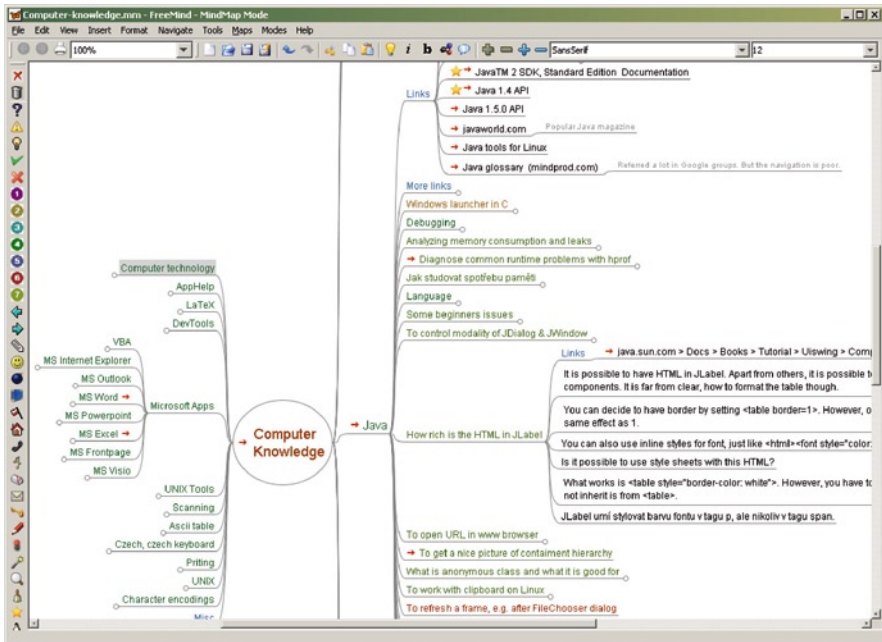


Fig. 2.3 Screenshot of a free mind mapping software called FreeMind (http://freemind.sourceforge.net/wiki/index.php/Main_Page)

Language: Mind Map

Mind Map were originally developed to support more efficient learning and evolved to a management technique used by numerous companies (Buzan 1974). Mind Map provides information about a topic that is structured in a tree (see Fig. 2.3 for example). Mind maps are used to generate, visualize, structure, and classify ideas, and as an aid in study, organization, problem solving, decision making, and writing.

Example: Information Ontology of Architectural Design

Bouattour et al. (2005) propose also a new set of concepts for information ontologies adapted to architectural design. These concepts could be seen as an upper layer of IFC classes (see section “Example: Industry Foundation Classes” (Ferreira da Silva and Cutting-Decelle 2005, p. 9)). Their information ontology is composed of actors, objects, activities and documents. All these components are in relation during the cooperative process of design building. Thus it is preferable to follow the decisions taken by each actor to understand the project development, to save time and to avoid errors. Their information ontology presents the state of architectural design components by following the decision process of each actor about this component.

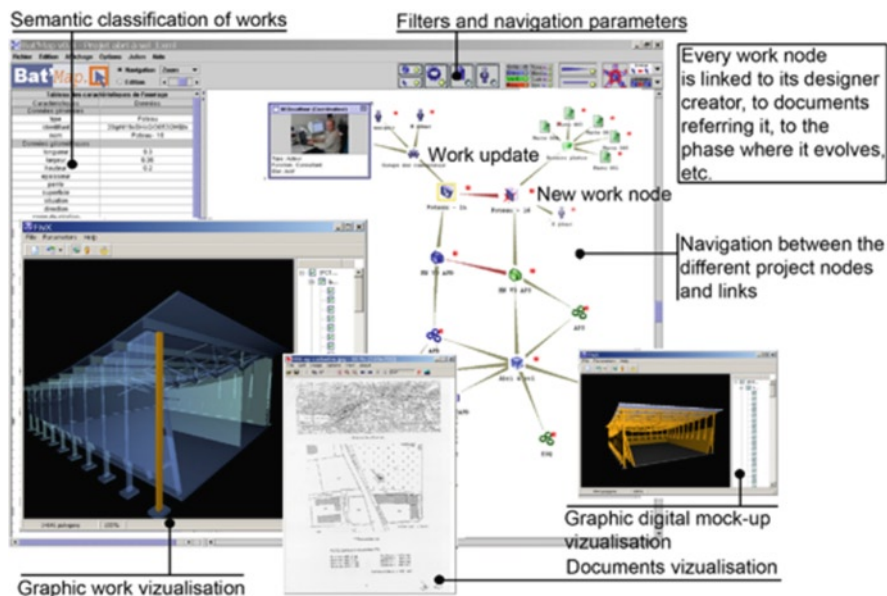


Fig. 2.4 Information ontology about architectural project

The information ontology representing the current state of an architectural project is composed of instantiation of their concepts. These ontologies are implemented in information system in order to compute some 3D representations of the building called mock up. These mock-ups synthesize the evolution of the project. This work is still in development, Bouattour et al. (2007) presents an on-going research aimed at computer-aided cooperative design for architectural project (Fig. 2.4).

Example: Information Ontology of Urban Planning

Kaza and Hopkins (2007) presents a set of concepts to formalize information ontologies used during urban planning process. Their information ontologies show the different alternatives of a decision in a plan. Plans could present effective decisions, alternative decisions and realizations in order to facilitate the communication between several actors. Moreover this type of plans can help stakeholders during their decision process in order to have a general overview of the city evolution. All these concepts (decisions, alternative, actors, etc.) and their instances compose an information ontology of urban planning (Fig. 2.5).

In this example the information ontology does not look like a Mind Map but it still uses a visual language similar to that used in a plan. This type of information ontology focuses on the location of the concept instance not on their internal structure description.

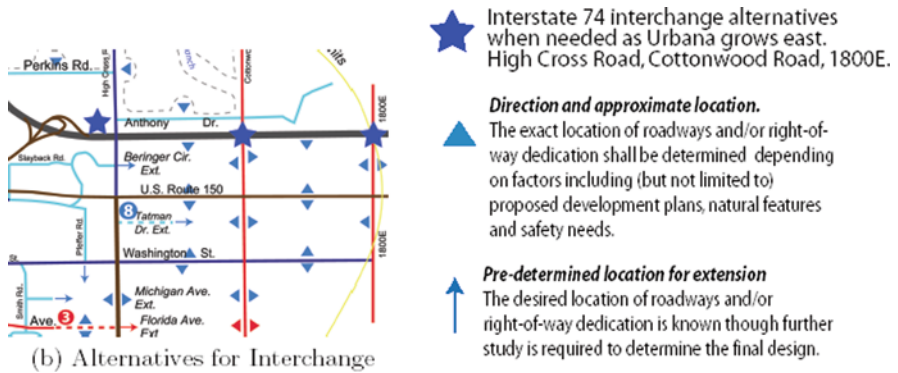


Fig. 2.5 Information ontology about urban planning process

Example: Information Ontology of Construction Project

Lee and McMeel (2007) propose to build an information ontology in order to ease the communication between the different actor groups involved in a construction project. These information ontologies represent some general patterns that have to be modified in order to resolve the specific problem of the construction project. The first stage of problem solving is to understand the language convention of each actor group based on the ontology element. Then negotiation and collaborative works can begin to find the appropriate solution of the construction problem. This type of ontology has to be heavy adaptable and modifiable.

2.2.1.2 Linguistic/Terminological Ontologies

Linguistic ontologies can be glossaries, dictionaries, controlled vocabularies, taxonomies, folksonomies, thesaurii, or lexical databases. As shown in Fig. 2.6 this type of ontology mainly focuses on terms and their relationships.

Unfortunately, terms are ambiguous. A concept can be referenced by several terms (for example: “computer science”, “computing”, “information technology” are synonyms) and a term can reference several concepts (for example the term “bank” can be used to reference a “river bank” or a “commercial bank”). The roles of linguistic ontologies are twofold: The first one is to present and define the vocabulary used. This is achieved by a dictionary for example which list all the terms actually used in language. Secondly, linguistic ontology is the result of a terminology agreement between a users’ community. This agreement defines which term is used to represent a concept in order to avoid ambiguity. This process is called vocabulary normalization. When a concept could be described by two synonym terms, the normalization process selects one of those to be the preferred label of the concept. It means that in Fig. 2.6 the cardinality of the hasLabel and hasID relationship is changed

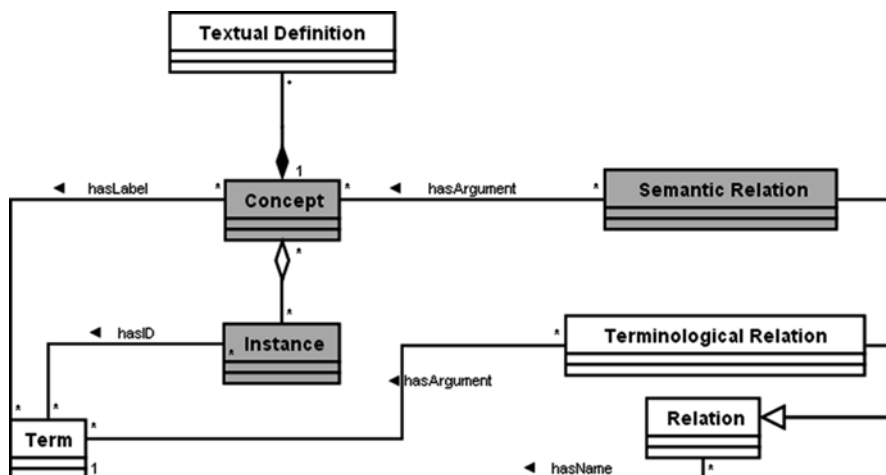


Fig. 2.6 UML schema of linguistic ontology components and their relationships

from * to 1 compared to Fig. 2.1. Taxonomy and thesaurus organized their normalized vocabulary so that the a priori relationships between concepts are made explicit. That is the reasons why in Fig. 2.6 concept and semantic relation are in grey to express that some linguistic ontologies try to explicit these components. Unfortunately the distinction between concepts and their instances are not taken in account: Instances are considered like concepts. A thesaurus has three basic relationships among terms: equivalence, hierarchical and associative. Let us point out that the last two relations hide several semantic relations. Associative relation between two terms means that there exists a semantic link between concepts labeled by these terms but no information is given on this semantic link. Hierarchical relation between two terms can hide an “instance of” relation between a concept and one of its instances (in grey in Fig. 2.6), a “specialization” relation between two concepts, a “part of” relation between concepts and so on. More information on thesaurus development are available in (ISO 2788 and ISO 5964).

Now we describe two languages that can be used to describe this type of ontologies: SKOS is used to define thesaurii and RDF is used the defined web metadata. Next we present four different thesaurii belonging to different domains: urban planning, environmental domain and cultural heritage; followed by a taxonomy used in architectural design.

Language: Simple Knowledge Organization System (SKOS)

Simple Knowledge Organization System (SKOS) is a semantic web activity proposed by the W3C. They are developing specifications and standards based on XML to support the use of knowledge organization systems such as thesauri, classification schemes, subject heading systems and taxonomies within the framework of the Semantic Web [see <http://www.w3.org/2004/02/skos/intro> for more details].

Language: Resource Description Framework (RDF)

The Resource Description Framework (RDF) is a general-purpose language for representing information in the Web. RDF is a recommendation from the W3C for creating meta-data structures that define data on the Web. RDF is used to improve searching and navigation for Semantic Web search engine (Web 3.0 applications).

RDF is implemented in XML. RDF is composed of Triples: (1) the subject (the web page), (2) a property or predicate (an attribute name) and (3) an object (the actual value of the attribute for the web page).

1. The subject is a resource. Resource is anything that can have a Unique Resource Identifier (URI); this includes all the world's web pages, as well as individual elements of an XML document.
2. The property is a resource that has a name. For example the Dublin Core Metadata Initiative propose to use the name "dc:creator" to represent the author property. Property can be associated to a property type defined in an RDF Schema (RDFS). RDFS defined a RDF vocabulary composed of property type and resource type.
3. The object can be a URI, a literal (a string of character representing a number, a date, a noun etc.) or a blank node.

For example, the triple (1) <http://www.textuality.com/RDF/Why.html> (2) dc:creator (3) "Tim Bray" means "The Author of <http://www.textuality.com/RDF/Why.html> is Tim Bray" [see <http://www.w3.org/RDF/> for more details].

Example: URBAMET Thesaurus. Urban Planning, Housing and Construction News and Records

URBAMET is the French library databank on urban development, town planning, housing and accommodation, architecture, public facilities, transport, local authorities etc. Since the creation of the data bank in 1986, the hierarchical organization of all these topics gave place to the construction of thesaurus URBAMET. The thesaurus is accessible in French, Spanish and English. A study of this thesaurus is presented in Chap. 10.

Example: GEMET Thesaurus

GEMET, the General Multilingual Environmental Thesaurus is the reference vocabulary of the European Environment Agency (EEA) and its Network (Eionet). It has been developed as an indexing, retrieval and control tool for the EEA.

GEMET was conceived as a "general" thesaurus, aimed to define a common general language, a core of general terminology for the environment. The language used in GEMET are: Basque, Bulgarian, Czech, Dutch, Danish, English, Estonian, Finnish, French, German, Greek, Hungarian, Italian, Norwegian, Polish, Portuguese, Russian, Slovenian, Slovak, Spanish, Swedish etc. [see <http://www.eionet.europa.eu/gemet> for more detail] (Fig. 2.7).

Thematic Listings INSPIRE Spatial Data Themes Alphabetic Listings Hierarchical Listings Search Thesaurus	
Select language: ar bg cs da de el en en-US es et eu fi fr hu it it lv mt nl no pl pt ro ru sk sl sv	
Themes list	
administration	legislation
agriculture	materials
air	military aspects
animal husbandry	natural areas, landscape, ecosystems
biology	natural dynamics
building	noise, vibrations
chemistry	physics
climate	pollution
disasters, accidents, risk	radiations
economics	research
energy	resources
environmental policy	social aspects, population
fishery	soil
food, drinking water	space
forestry	tourism
general	trade, services
geography	transport
human health	urban environment, urban stress
industry	waste
information	water

Fig. 2.7 The theme list of the GEMET thesaurus

Example: Agrovoc Thesaurus

In environmental domain, the well-known AGROVOC thesaurus is used to develop the Agricultural Ontology Service (AOS) project (AGROVOC). AGROVOC is a multilingual thesaurus designed to cover the terminology of all subject fields in agriculture, forestry, fisheries, food and several other environmental domains (environmental quality, pollution, etc.). As presented in (AGROVOC), “it consists of words or expressions (terms), in different languages and organized in relationships (e.g. ‘broader’, ‘narrower’, and ‘related’), used to identify or search resources”. AGROVOC was developed by the FAO and the Commission of the European Communities, in the early 1980s. It is an excellent example of linguistic ontology resulting of a terminology agreement between a community. The terms of AGROVOC can be used to reference document contents (Wildemann et al. 2004) or to find the similarity degree between several words corresponding to the same idea. AGROVOC is available in the following languages: English, French, Spanish, Arabic, Chinese, Portuguese, Czech, Thai, Japanese, Lao Hungarian, Slovak, German, Italian, Polish, Farsi (Persian), Hindi, Telegu, Moldavian [see http://www.fao.org/aims/tools_thes.jsp for more detail].

Example: HEREIN Thesaurus

The European Heritage Network is an information system gathering governmental services in charge of heritage protection within the Council of Europe. The HEREIN project focuses is on cultural heritage, particularly on architectural and on archaeological heritage. The multilingual thesaurus attached to the HEREIN project intends to offer a terminological standard for national policies dealing with architectural

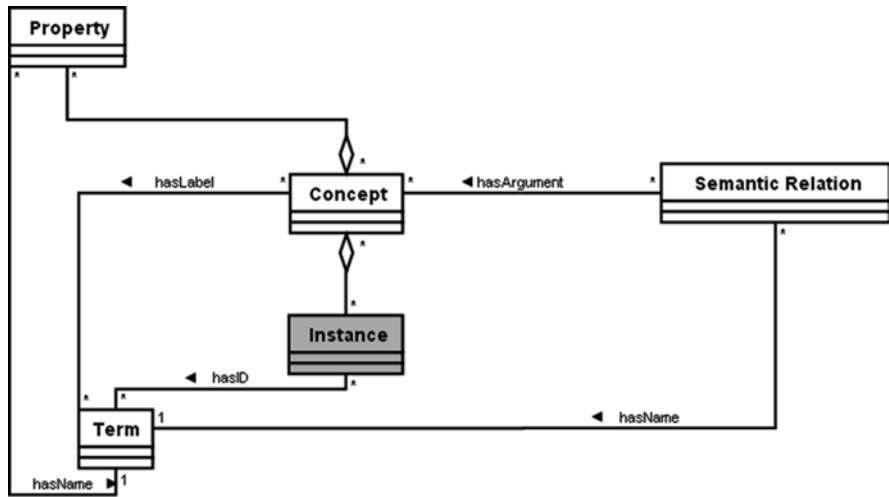


Fig. 2.8 UML schema of software ontology components and their relationships

and archaeological heritage [see <http://www.european-heritage.net/sdx/herein/index.xsp> for more detail]. This thesaurus is described in Chap. 15.

Example: DesignScape Project

The research developed in the project DesignScape focuses on the modeling of the different steps of the architectural design (Kim and Kim 2007). The works formalize the typical building design process by a linguistic ontology. More precisely, the ontology is a taxonomy describing the relationships between different activities related to architectural design. The main basis activities modeled are: pre-design, site analysis, schematic design, space zoning, site zoning, objectives definition, analysis, synthesis, evaluation. Numerous concepts around the architectural design activity are represented in the considered ontology.

2.2.1.3 Software Ontologies

Software ontologies (or software implementation driven ontologies) provide conceptual schemata whose main focus is normally on data storage and data manipulation, and are used for software development activities, with the goal of guaranteeing data consistency. As shown in Fig. 2.8, a concept is composed of a set of properties; all concepts are also defined thanks to each other’s by the relations they have. These relations are also associated to constraints. At execution time, data are stored in the properties of object, that is to say an instance of a concept. Thus, data could be processed in various treatments (called methods). Nevertheless, software

ontologies goal is not to describe particular instances during execution time; that is the reason why instance is colored in grey in Fig. 2.8.

Software ontologies are normally defined with conceptual modeling languages used in software and database engineering. These languages are used during software design procedure: for example Entity-Relationship Model language or Object Model Language. The next section presents the most well known one called UML. UML presentation will be followed by one example of software ontology¹ used in building construction.

Language: Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a standard used mainly for modeling software and information systems. UML is a graphical language for visualizing, specifying and constructing any parts of software components. UML is a semi-formal formalism, because the official document defining the semantics of UML is mainly composed of informal descriptions in English (OMG 2003). Thus, UML is not sufficient to represent all the details required by complex reasoning processes (Cranefield 2001) like: deducing new knowledge, compute the logical correctness of a formal ontology, etc. UML propose several diagrams, the ones used to specify software ontology are UML class diagram and UML object diagram. Figures 2.1, 2.2, 2.6, 2.8, 2.9 are examples of UML class diagram.

Example: Industry Foundation Classes

The Industry Foundation Classes (IFC) has been progressively developed by the International Alliance for Interoperability (IAI) since 1995. There have been several releases of the model that have been implemented. The IFC is a response to interoperability requirements within building construction by a significantly large group of industry practitioners including government and other statutory bodies, clients, consultants and contractors together with a substantial number of software vendors. The primary target of the IFC Model is the interoperability among software applications within the building and construction market sector (Ferreira da Silva and Cutting-Decelle 2005). IFC classes are therefore defined according to the scope and the abstraction level of software systems dealing with building and construction specific content. Such a model has been primarily developed to enable the exchange and sharing of Building Information Models (BIM) to increase the productiveness of design, construction, and maintenance operations within the life cycle of buildings. The IFC model therefore describes an object model with concepts (classes), relations (as direct associations or objectified relationships), and properties (or attributes).

¹Software ontology can be also name semi formal ontology in the literature because UML is a semi-formal formalism.

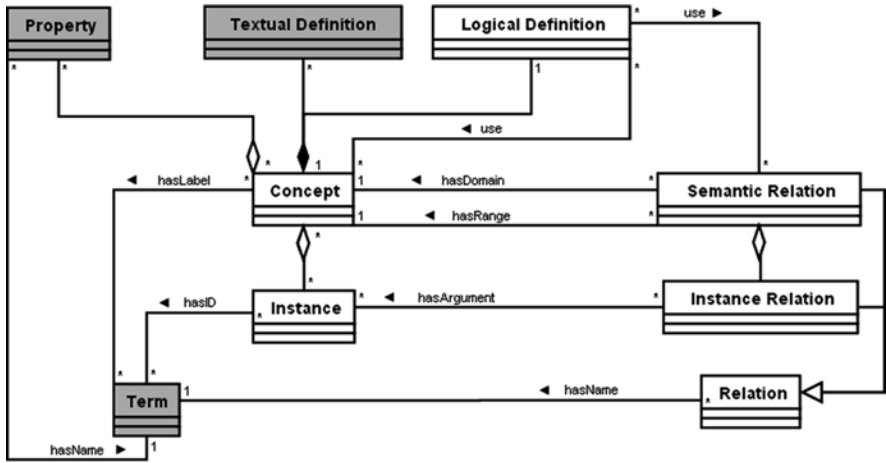


Fig. 2.9 UML schema of formal ontology components and their relationships

IFC classes are first built in the express language, and now an XML version is available. They are now widely accepted by industry and major Computer Aided Design software systems support IFC classes for file based exchanges with planning tools and cost evaluation applications. The IFC standard is studied in several chapters of this book, especially in Chap. 8.

2.2.1.4 Formal Ontologies

Formal ontologies require a clear semantics for the language used to define the concept, clear motivations for the adopted distinctions between concepts as well as strict rules about how to define concepts and relationships. This is obtained by using formal logic (usually first order logic or Description Logic) where the meaning of the concept is guaranteed by formal semantics (Borgo 2004). As you can see in Fig. 2.9, this ontology type is the only one that contains logical definition.

For example, Knowledge Bases (KB) are formal systems that capture the meaning of the adopted vocabulary via logical definitions. The logical definition of a concept is composed of one or more logical formulae. A logical formula (or axiom) is a combination of concepts and semantic relations. A KB contains more expressive components than a conceptual schema (Notice in Fig. 2.9 that formal ontology has all the components of software one). The purpose is not simply retrieval and storage of data but reasoning. Compared to software ontology, data are not associated to method in order to make some calculation; data are stored in property only to be retrieved (That is the reason why property is in grey in Fig. 2.9). For example, Fig. 2.10 presents an ontology about urban development and civil engineering, which defined knowledge related to getting urbanism authorizations for new buildings.

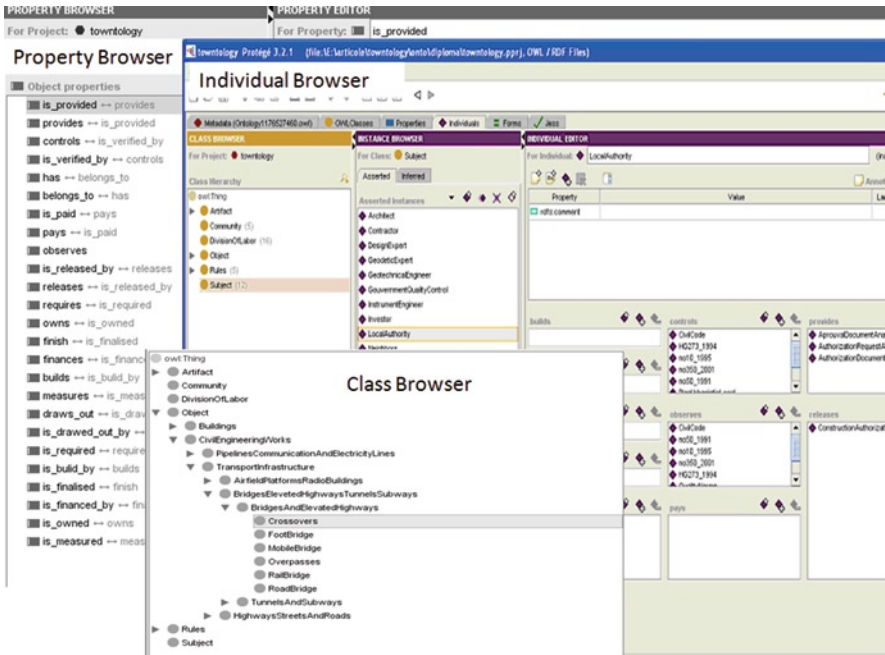


Fig. 2.10 Protégé knowledge base ontology about urban development and civil engineering ontology

This Knowledge Base is defined with the Protégé system (Protégé 2005). Thanks to these formal definitions and rules, the inference engine can enter into a dialog with a user (see Chap. 9). Formal ontology does not focus on term and textual definition even if they could be defined in the ontology. Terms are only used as symbol in order to help user during the manipulation of logical formula.

There exist different formal languages used to describe formal ontology like Description Logics (DL), Conceptual Graphs (CG), First Order Logic (FOL), etc. We chose to present OWL, the standard recommended by W3C. OWL presentation is followed by three examples of formal ontologies belonging to urban planning field, architecture domain and pervasive environment.

Language: Web Ontology Language (OWL)

The OWL Web Ontology Language is a standard recommended by the W3C. It is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. The OWL is intended to provide a language that can be used to describe concepts and

relations between them that are inherent in Web documents and applications. OWL language is used for:

1. formalize a domain by defining concepts called classes and properties of those classes,
2. define instances called individuals and assert properties about them,
3. reason about these classes and individuals to the degree permitted by the formal semantics of the OWL language.

The OWL language provides three increasingly expressive sublanguages designed for use by specific communities of implementers and users.

- OWL Lite supports those users primarily needing a classification hierarchy and simple constraint features. For example, while OWL Lite supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and provide a quick migration path for thesauri and other taxonomies.
- OWL DL supports those users who want the maximum expressiveness without losing computational completeness (all entailments are guaranteed to be computed) and decidability (all computations will finish in finite time) of reasoning systems. OWL DL includes all OWL language constructs with restrictions such as type separation (a class can not also be an individual or a property, a property can not also be an individual or class). OWL DL is so named due to its correspondence with Description Logics, a field of research that has studied a particular decidable fragment of first order logic. OWL DL was designed to support the existing Description Logic business segment and has desirable computational properties for reasoning systems.
- OWL Full is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. Another significant difference from OWL DL is that a `owl:DatatypeProperty` can be marked as an `owl:InverseFunctionalProperty`. OWL Full allows an ontology to augment the meaning of the pre-defined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support every feature of OWL Full.

Each of these sublanguages is an extension of its simpler predecessor, both in what can be legally expressed and in what can be validly concluded. The following set of relations hold. Their inverses do not [see <http://www.w3.org/TR/owl-features/> for more details].

Example: Formal Ontology About Urban Development and Civil Engineering

Chapter 9 of this book contains a description of an expert system able to dialog with a user in order to inform him about which document is necessary to have the authorization to construct a new building. The goal is not only to give a list of

documents but to explain where and how to find this document and where to send this document... Moreover the process could be an incremental process to ask for this document you will need this one and so on... This expert system consists of a formal ontology about urban development and civil engineering and an the rule-based inference engine Jess.²

This ontology contains concepts called classes, instances called individuals and relations called properties. For example, the “Subject” class has 12 instances (see Fig. 2.10). One of these, the “LocalAuthority” instance has several relations (“provides”, “releases”, “controls”, etc.) with other individuals.

Example: Formal Ontology for the Korean Architectural Domain

A research project has been initiated in South Korea to model an ontology for the Korean architectural domain (Kim 2005). The author has built the formal description of the ontology in using OWL and Protégé. A prototype based on this ontology has been developed to help learn the History of the main Korean historical buildings.

The goal is to precisely and formally model information related to monuments (temples, towers, famous places, etc.) and their history. The main classes of the ontology are:

- Buildings,
- Architectural Styles,
- Artifacts e.g. important monuments associated to buildings; artifacts are linked to buildings by the relationship “belong to”
- History; this class has three subclasses (1) Event, (2) Fact, (3) Story i.e. legend and non-verified information,
- People,
- Media (image, sound, text, etc.).

For instance, the ontology can be used to model that two towers Dabo-Pagoda and Seokga-Pagoda (i.e. two instances of Artifacts) belong to the Temple Bulgulsa (i.e. an instance of Building). In this example, historical Events associated to the temple can be “thievery” or “fire”. An example of Facts is “KimDaesung has constructed the temple”. In the ontology, the person KimDaesung is an instance of People. All the media providing significant information are instances of the class Media.

Example: CoBra Ontology

In another context, the CoBra ontology has been defined in order to facilitate the pervasive computing environment (Chen et al. 2003). The ontology has been modeled in OWL to enable reasoning about knowledge.

²<http://www.jessrules.com/>

In a pervasive environment, different intelligent agents have specific tasks and they can communicate each other thanks to a common network. Agents must exchange their knowledge and cooperate together to understand a local context and reach their goals. This requires to reason with contexts and to help agents maintaining consistent contextual knowledge and cooperating. The authors implemented an intelligent meeting room system on the campus of an university. The ontology (consisted of 17 classes and 32 properties definitions) defines some of the common relationships and attributes that are associated with people, places and activities in an intelligent space. When a person enters into a conference room, the system detects his presence and acquires situational information from heterogeneous sources such as the Web, corporate databases, etc. The situational information may concern data about the person, schedule, time, technical characteristics of the devices installed in the room, etc. Then, the system deduces the role and the intention of the person by reasoning with the context and by interpreting situational information (e.g. he is expected to present something in this room at that time). The system informs other agents in order to help this person e.g. the projector agent can obtain automatically the presentation from system and run the slideshow.

This use of formal ontologies could have several direct applications in the management of urban areas. Different agents could be implemented in the infrastructure of the city (public transport, automatic distributor teller, etc.). Each agent will deduce what people need thanks to contextual information and a formal ontology.

2.2.2 Classification Based on the Scope of the Ontology, or on the Domain Granularity

Figure 2.11 presents the second classification based on the scope of the objects described by the ontology. For instance, the scope of a local ontology is narrower than the scope of a domain ontology; domain ontologies have more specific concepts than core reference ontologies, which contains the fundamental concept of a domain. Foundational ontologies can be viewed as meta ontologies that describe the top level concepts or primitives used to define others ontologies. Finally, general ontologies are not dedicated to a specific domain thus its concepts can be as general as those of core reference ontologies.

2.2.2.1 Local Ontologies/Application Ontologies

Local or application ontologies are specializations of domain ontologies where there could be no consensus or knowledge sharing. This type of ontology represents the particular model of a domain according to a single viewpoint of a user or a developer.

Fonseca et al. (2000) present this kind of ontology as a combination of domain ontology and task ontology in order to fulfill the specific purpose of an application.

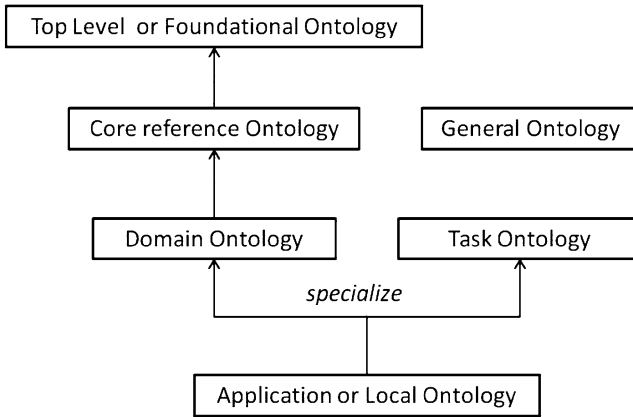


Fig. 2.11 Ontology classification based on domain scope

The task ontology contains knowledge to achieve a task, on the other hand the domain ontology describes the knowledge where the task is applied.

2.2.2.2 Domain Ontologies

Domain ontology is only applicable to a domain with a specific view point. That is to say that this viewpoint defines how a group of users conceptualize and visualize some specific phenomenon. This domain ontology could be linked to a specific application: electric network management system for example.

Example: Urban Sprawl Ontology

Cagiloni and Rabino (2007) use the Lowry Model of the city in order to have a simplified view of the urban sprawl phenomenon. The Lowry Model is a simplified model of the city that modeled the relation between transportation and land use. This model has a mathematical formulation taking as input values the employment, the population, the residential sector, the travel cost etc.. This ontology is a domain ontology, it is applicable only on urban morphological evolution.

2.2.2.3 Core Reference Ontologies

Core reference ontology is a standard used by different group of users. This type of ontology is linked to a domain but it integrates different viewpoints related to specific group of users. This type of ontology is the result of the integration of several

domain ontologies. A core reference ontology is often built to catch the central concepts and relations of the domain.

Example: Hydrontology

Vilches Blazquez et al. (2007) present the development of a core reference ontology untitled hydrontology describing hydrographic features. They gather different sources of information. These sources are chosen based on their reliability that is to say they come from well-known institution. Their goal is to harmonize all the different representations of hydrographic phenomenon in order to propose a standard. Hydrontology is presented in the Chap. 6.

Example: CityGML

CityGML is an OpenGIS® Encoding Standard for the representation, storage and exchange of virtual 3D city and landscape models. CityGML is implemented as an application schema of the Geography Markup Language version 3.1.1 (GML3).

CityGML models both complex and georeferenced 3D vector data along with the semantics associated with the data. Indeed, CityGML defines the classes and relations for representing the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantical, and appearance properties. Thus CityGML can be seen as a core reference ontology dealing with 3D City Model. Nevertheless, CityGML is only appropriate for visualization purpose and it is not sufficient for touristic or environmental application. For these specific domain areas, CityGML also provides an extension mechanism to enrich the data with identifiable features. Thus, to our point of view, CityCML is a core reference ontology based on the GML foundational ontology and it can be derived to build domain ontology [see <http://www.citygml.org/> for more details]. CityGML is also discussed in Chap. 7.

2.2.2.4 General Ontologies

General ontologies are not dedicated to a specific domain or fields. They contain general knowledge of a huge area.

Example: OpenCyc Ontology

Cyc technology is a general knowledge base and commonsense reasoning engine. The entire Cyc ontology containing hundreds of thousands of terms, along with millions of assertions relating the terms to each other, forming an general formal

ontology whose domain is all of human consensus reality. The OpenCyc ontology is available in OWL format [see <http://www.opencyc.org/> for more detail].

2.2.2.5 Foundational Ontologies/Top Level Ontologies/Upper Level Ontologies

Foundational or top level ontologies are generic ontologies applicable to various domains. They define basic notions like objects, relations, events, processes and so on. All consistent ontology has a foundational ontology. Foundational ontology can be compared to the meta model of a conceptual schema (Fonseca et al. 2003). The most well known foundational ontology are the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)³ and the Basic Formal Ontology⁴ (BFO). These two ontologies are formal and propose a different logical theory for representation of world assumption. Thus, domain or core reference ontologies based on the same foundational ontology can be more easily integrated. For example, Fonseca et al. (2006) describes a top level ontology of geographic objects and a similarity measure to evaluate the interoperability of domain ontologies based on this top level ontology.

Example: Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)

Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) contains abstract concepts aimed at generalizing the set of concepts that we may encounter in different domains (Masolo et al. 2003).

DOLCE is a formal ontology of particulars, in the sense that its domain of discourse is restricted to them. The fundamental ontological distinction between universals and particulars can be informally understood by taking the relation of instantiation as a primitive: Particulars are entities which have no instance; universals are entities that can have instances. Properties and relations are usually considered as universals.

DOLCE describes particulars that can be physical object (endurant), events (perdurant), quality (quality) and quale (quality value).

- Endurants are entities enduring in time. Within endurants, physical objects are distinguished from non-physical objects, since only the former possess direct spatial qualities. The domain of non-physical objects covers social entities and cognitive entities.
- Perdurants are entities that happen in time and in which endurants participate. Among perdurants, one defines actions that are intentionally accomplished, i.e. controlled by an agent.

³See <http://www.loa-cnr.it/DOLCE.html> for more details.

⁴See <http://www.ifomis.org/bfo> for more details.

- Endurants and perdurants have inherent properties (qualities) that we perceive and/or measure.
- These qualities take a value (quale) within regions of values which are abstract.

In this context spatiality is expressed by a spatial quality which quale belongs to a spatial region at a point in time (Bateman and Farrar 2004).

For example, to express the fact that this paper (paper#1) is a Physical Endurant which has a Physical Quality (spatialQuality#1), we can say that there exists a relation QT between these two elements. This is formally expressed by the formula:

$$\text{PhysicalEndurant}(\text{paper}\#1) \wedge \text{PhysicalQuality}(\text{spatialQuality}\#1) \wedge \text{QT}(\text{paper}\#1, \text{spatialQuality}\#1).$$

To define the value of this spatial quality we need a Physical Region (location#1) which can be a point. This Physical Region is part of a space region (space region) and they are linked at a point in time t by a quale relationship QL.

This is expressed by the formula:

$$\text{PhysicalRegion}(\text{location}\#1) \wedge \text{PartOf}(\text{location}\#1, \text{space region}) \wedge \text{QL}(\text{location}\#1, \text{spatialQuality}\#1, t).$$

Example: Socio Cultural Ontology

Trausan-Matu (2007) presents the top level concept of a socio cultural ontology. These top level concepts come from the Activity theory of Engestrom. Yrjö Engeström's (1987) theory emphasizes categories (subjects, objects, and communities) and mediators (general artifacts, social rules and division of labor). Thus we could say that these six general concepts composed the foundational ontology of socio cultural ontology.

Example: Geography Markup Language GML

The Geography Markup Language (GML) is an OpenGIS® Encoding Standard for the representation, storage and exchange of geographical features. GML serves as a modeling language for geographic systems as well as an open interchange format for geographic transactions on the Internet. The concept of feature in GML is a very general one and includes not only conventional “vector” or discrete objects, but also coverage and sensor data.

To a technical point of view, GML is an XML grammar proposed by the Open Geospatial Consortium (OGC) to describe generic geographic data sets that contain points, lines and polygons. Thus GML contains a foundational ontology of geographic features. Moreover, GML can be extended to define core reference ontology called community-specific application schemas like CityGML.⁵ Using application schemas, users can refer to roads, highways, and bridges instead of points, lines and polygons.

⁵see <http://www.opengeospatial.org/standards/gml> and http://en.wikipedia.org/wiki/Geography_Markup_Language for more details.

2.3 Different Ontology Design Approaches

Several methodologies for ontology engineering are proposed to design ontologies. The most complete ones are Methontology (Gomez-Perez et al. 2004) and On-to-knowledge (Sure et al. 2003). Nevertheless, this research area is still in development; see for example the NeON project [<http://www.neon-project.org/web-content/>]. All these methodologies are composed of several activities. The development process is not a linear process but a refinement one where each activity can be repeated several times. Among all the activities the most important are:

- Ontology specification
- Knowledge acquisition
- Conceptualization
- Formalization
- Implementation
- Evaluation
- Maintenance
- Documentation

2.3.1 Classification Based on Taxonomy Construction Direction

The conceptualization activity is composed of several tasks. One of them is the construction of the concept taxonomy.

To build the taxonomy of concepts, several approaches have been opposed in literature (Gandon 2002):

- *Bottom-Up approach.*
- *Top-Down approach.*
- *Middle-Out approach*

2.3.1.1 Bottom Up Approaches

Bottom-Up approaches start from the most specific concepts and build a structure by generalization; the ontology is built by determining first the low taxonomic level concepts and by generalizing them. This approach is prone to provide tailored and specific ontologies with fine detail grain concepts (Gandon 2002).

Example: Spatial Database Ontology

Chaidron et al. (2007) investigate a bottom up approach of local ontology construction. Their goal is to define a local ontology describing objects stored in a spatial database.

Thus, this local ontology is used as a DB documentation. Moreover, it facilitates the database reengineering by differentiating the attended use and the effective use of object type. This work is presented in Chap. 6.

Example: Urban Network Ontology

Nogueras-Iso et al. (2007) propose an automatic bottom up method of ontology construction based on Formal Concept Analysis approach. Their goal is to analyze several databases of street type and personal or land addresses in order to build a core reference ontology of street types. Thus this ontology is the result of merging different street categorizations. Each database has its own street categorization. This core reference ontology could be used afterward to query all these databases together in a general system (see chapter object interoperability).

Example: The Phenomen Ontology

Gomez-Perez et al. (2008) propose to study a cartographic database of the National Geographic Institute of Spain to build automatically a first draft of a domain ontology about geographic feature types. They used a combination of criteria to build taxonomy of concepts. First of all, the code of each feature instance stores a three level classification. Secondly, they notice that the several features names can begin with a common lexical part (for example: “highway”, “divided highway”, “toll highway”). The common part “highway” is the super class of the others feature types.

2.3.1.2 Top Down Approaches

Top-Down approaches start from the most generic concept and build a structure by specialization; the ontology is built by determining first the top concepts and by specializing them. The top concepts can be chose in a foundational ontology These approaches are prone to the reuse of ontologies and inclusion of high level philosophical considerations which can be very interesting for coherence maintenance (Gandon 2002).

Example: Socio Cultural Ontology

Human activity creates some geographical object like state, city, administrative border. These types of geographical objects are opposed to physical object like mountains, rivers. Trausan-Matu (2007) describe urban development as the consequence of human activity. He uses a top down approach to develop its socio cultural ontology. This ontology explains the human interaction inherent in urban development. He adapts the John Sowa’s methodology of classification (Sowa 1999) to the activity theory of Engestrom. This ontology is presented in the Chap. 9.

Example: Urban Morphological Process Ontology

Camacho-Hubner and Golay (2007) propose to use John Sowa's methodology to develop an ontology of urban morphological change. Their goal is to explain the change of urban morphology by studying 3 aspects: time, historical context and morphological process.

2.3.1.3 Middle Out Approaches

Middle-Out approaches identify central concepts in each area/domain identified; core concepts are identified and then generalized and specialized to complete the ontology. This approach is prone to encourage emergence of thematic fields and to enhance modularity and stability of the result (Gandon 2002).

Example: Hydrotonology

Vilches Blazquez et al. (2007) presents an experiment in building a core reference ontology of hydrographic features. They use the Methodology ontology development method based on middle out approach. First they build a dictionary of most important terms. These terms enable to start the development of the concept hierarchy using four taxonomic relations: subclass of, disjoint-decomposition, exhaustive decomposition and partition. A dictionary was drawn up and used to validate the correctness of the taxonomy. The ontology was enriched by adding ad-hoc relationship between ontology concepts. The last step was the attribute specification for every concept. For more detail see Chap. 6.

2.3.2 Classification Based on the Type of Sources

As shown previously ontology design contains the activity of knowledge acquisition. This activity is based on elicitation techniques. The task of knowledge elicitation from resources gave rise a new research field called ontology learning. The second ontology design classification is made according to the type of information sources used to extract knowledge. You will find a more precise description on all the possible techniques to build an ontology from different resources in Chap. 10.

2.3.2.1 Based on Text

Texts contain unstructured data not meaningful for a computer system. Nevertheless textual corpus is a huge source of information. Texts can be used to extract terms thanks to Natural Language Processing technique (lots of parser are available like tree tagger).

Then statistical techniques like co-occurrence evaluation can be applied in order to extract the most important terms appearing in the corpus or the most important couple of terms in order to detect relation between terms. Linguistic pattern can also be extracted thanks to statistical technique. Those patterns identify specific semantic relations thanks to verb extraction.

For example, Aussenac Gilles et al. (2000) presents an experiment of corpus analysis in order to help the knowledge engineer during the design of domain ontology. The design process is still manual but corpus analysis tools minimize time and efforts spent in knowledge extraction.

Buitelaar et al. (2005) present an overview of different techniques applied on text in order to extract knowledge. Knowledge can be: terms, synonyms and multilingual variants, concepts (concept can be identified as a set of instances or a cluster of terms or a textual definition), concept hierarchies, non-hierarchical relations and rules. These elicitation techniques come from Information Retrieval methods, Natural Language Processing tool, text mining methods or statistics.

Mounce et al. (2009) presents an example of semi automatic creation of ontology in the water field. They use the ontology learning tool Text2Onto applied on corpora of documentation about water management.

2.3.2.2 Based on Thesaurus

Thesaurii are linguistic ontologies that can be used to extract a first draft of software or formal ontology.

For example Lacasta et al. (2007) presents an experiment in using several multilingual thesaurus (AGROVOC, EUROVOC, GEMET, UNESCO, URBISOC thesaurus) in order to build a first draft of a domain ontology in urbanism. Their goal is to extract concepts and semantic relations from terms and linguistic relations. Notice that this work is detailed in the Chap. 10. The process is composed of several steps:

- Thesaurii are transformed in the same format.
- Terms of different thesaurus are gathered when they share at least a common synonym. Thanks to these clustering techniques a concept is considered as a cluster of terms.
- The probability of concept is proportional to the number of synonyms shared by different thesaurii.
- Semantic relations are extracted from linguistic relations between concepts contained in different thesaurii.
- Semantic relation probability is proportional to the number of linguistic relations between terms contained in different thesaurii.

2.3.2.3 Based on Relational Database

Relational databases are valuable resources for software or formal ontology learning. Due to the structural nature of database, a better accuracy of ontology design

process can be expected than from textual corpora. In relational database two aspects can be explored to design ontology: the structure of the database and the data stored in the database. The structure can be used to extract concepts, semantic relations and properties. Li et al. (2005) and Astrova and Kalja (2008) works are based on rules to transform a well-formed relational database into an object oriented conceptual schema. Nevertheless the resulting ontology has a flat concept taxonomy. We need to extract more hierarchical relation from data. For example Nogueras-Iso et al. (2007) use the Formal Concept Analysis techniques applied on the data of different urban network databases to build a taxonomy of street feature types. Gomez-Perez et al. (2008) analyze the code and the name of geographic features of the National Geographic Institute of Spain to build a taxonomy of geographic feature types. Lammari et al. (2007) consider that partitioning of a database table on the basis of null values may reveal concept hierarchies. Cerbah (2008) proposes a tool to refine the concept hierarchy based on property representing type values.

2.3.2.4 Based on UML Diagrams

Some works propose a formal ontology design methodology based on UML models (Cranefield and Purvis 1999; Philippe 2003; Schreiber 2005; IBM 2006; Gasevic et al. 2006; Pinet et al. 2009).

One main advantage of UML is that is taught in many Departments of Universities, and it is widely used, even by non-computer scientists. UML is supported by several tools so designers can use them for describing their diagrams. Users and developers are likely to be familiar with UML notations than traditional formal ontology based languages. For all these reasons, UML could be viewed as a good candidate to model formal ontologies.

There are several common features between UML and formal Ontology-based languages (IBM 2006) but the main drawback of UML is its lack of formal semantics. Some researchers propose a mathematical model for UML (see for instance Breu et al. 1997). The work of Guizzardi (Guizzardi et al. 2002, 2004) concerns the development of different methodological tools (UML profiles, design patterns) in order to build an ontology using UML formally and correctly. Cranefield and Purvis show how a formal ontology can be built using UML and Object Constraint Language (OCL) (OMG 2006); the concepts are described by UML classes and constraints on concepts are described in OCL (Cranefield et al. 1999).

OCL is a textual language that might overcome the limitation of UML in the future. In the Object Management Group (OMG) specification of OCL (OMG 2006), an annex presents a first version of a formal semantic but currently, this annex does not describe how to make complex reasoning processes within OCL constraints.

An interesting methodology to develop an ontology is to capture (by using UML diagrams) consensual knowledge accepted by the experts. In a second step, the software ontology described in UML can be translated into a formal ontology (RDF, OWL, etc.). Then this produced formal ontology can be enriched in order to offer

new possibilities: produce reasoning, reach the requirement of the semantic Web, integrate several database schemas, etc.

For example, the works of Pinet et al. (2006) propose to start modeling a software ontology with a UML class diagram. After that, this UML specification is translated into OWL with Protégé and its UML Storage Backend Plug-In (Protégé). Then, additional specifications are defined with Description Logic in order to produce a formal ontology.

2.4 Conclusion

Ontologies have been used for the last decades for a set of tasks: improving communication between agents (human or software) or reusing data model or knowledge schema. All these tasks deal with interoperability issues and can be applied in different domains. Consequently, ontologies have evolved and several kinds of ontologies have been proposed.

We have presented several visions of ontology types and how to build them. Moreover we have described the main components of each type of ontology. Several examples have been provided in order to help understand the different uses of ontologies.

In the next chapter, we will show which types of system interoperability issues can be resolved by ontologies, and which types of ontologies have been used for this purpose.

References

- Astrova, I., Kalja, A.: Automatic transformation of SQL relational databases to OWL ontologies. In: WEBIST, vol. 2, pp. 131–136 (2008)
- Aussenac-Gilles, N., Biebow, B., Szulman, S.: Revisiting ontology design: a methodology based on corpus analysis. In: EKAW, Juan-les-Pins, pp. 172–188 (2000)
- Bateman, J., Farrar, S.: Towards a generic foundation for spatial ontology. In: Varzi, A.C. (ed.) Formal Ontology in Information Systems: Proceedings of the Third International Conference (FOIS-2004), Laure Vieu Publié par IOS Press (2004)
- Borgo, S.: Classifying (medical) ontologies. Tutorial for the Ontology Workshop at the Semantic Mining Summer School (2004)
- Borgo, S.: How formal ontology can help civil engineers. In: Teller, J., Lee, J., Roussey, C. (eds.) Ontologies for Urban Development: Interfacing Urban Information Systems. Studies in Computational Intelligence, vol. 61, pp. 143–156. University of Geneva, 6,7 Nov 2006. Springer Verlag (2007). ISBN 978-3-540-71975-
- Bouattour, O., Halin, G., Bignon, J.-C.: A Cooperative model using semantic works dedicated to architectural design. In: Proceedings of the 10ème Conférence – CAADRIA The Association for Computer Aided Architectural Design Research in Asia, New Delhi, 28–30 Avril, 2005, Publication primée: Woung CAADRIA Award 2005, (2005)
- Bouattour, O., Halin, G., Bignon, J.-C.: Management system for a virtual cooperative project. In: Proceedings of the eCAADe Conference, Frankfurt, Allemagne, Sept 2007

- Breu, R., Grosu, R., Huber, F., Rumpe, B., Schwerin, W.: Towards a precise semantics for object-oriented modeling techniques. In: Object-Oriented Technology, ECOOP'97 Workshop Reader (1997)
- Buitelaar, P., Cimiano, P., Magnini, B., et al.: Ontology learning from text: an overview. In: Buitelaar, P., Cimiano, P., Magnini, B. (eds.) *Ontology Learning from Text: Methods, Evaluation and Applications* *Frontiers in Artificial Intelligence and Applications Series*, vol. 123. IOS Press, Amsterdam (2005)
- Buzan, T.: *Use your head*. BBC Books, (1974)
- Caglioni, M., Rabino, G.: Theoretical approach to urban ontology: A contribution from urban system analysis. In: Teller, J., Lee, J., Roussey, C. (eds.) *Ontologies for Urban Development: Interfacing Urban Information Systems*. *Studies in Computational Intelligence*, vol. 61, pp. 143–156. University of Geneva 6,7 Nov 2006. Springer Verlag (2007). ISBN 978-3-540-71975-
- Cerbah, F.: “Learning highly structured semantic repositories from relational databases - RDBtoOnto tool”. In: *Proceedings of the 5th European Semantic Web Conference (ESWC 2008)*, Tenerife, Spain, June 2008
- Chaidron, C., Billen, R., Teller, J.: Investigating a bottom-up approach for extracting domain ontologies from urban databases. In: Teller, J., Lee, J., Roussey, C. (eds.) *Ontologies for Urban Development: Interfacing Urban Information Systems*. *Studies in Computational Intelligence*, vol. 61, pp. 143–156. University of Geneva 6,7 November 2006. Springer Verlag (2007). ISBN 978-3-540-71975-
- Chen, H., Finin, T., Joshi A.: An ontology for context-aware pervasive computing environments. *J. Knowl. Eng. Rev.* **18**(3), 197–207 (Sept 2003). Cambridge University Press, USA (2003). ISSN:0269-8889
- Cranefield, S., Purvis, M.: UML as an ontology modelling language. In: *Proceedings of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI-99)* (1999)
- Cranefield, S.: UML and the Semantic Web. In: *The International Semantic Web Working Symposium, Palo Alto*, (2001)
- Ferreira Da Silva, C., Cutting-Decelle, A.-F.: Industrial standards to structure the construction information. COST C21 Technical report no2. Available at <http://www.towntology.net/references.php> (2005)
- Fonseca, F., Câmara, G., Monteiro, A.M.: A framework for measuring the interoperability of geo-ontologies. *Spat. Cogn. Comput* **6**(4), 307–329 (2006)
- Fonseca, F., Davis, C., Camara, G.: Bridging ontologies and conceptual schemas in geographic applications development. *Geoinformatica* **7**(4), 355–378 (2003)
- Fonseca, F., Egenhofer, M., Davis, C., Borges, K.: Ontologies and knowledge sharing in Urban GIS. *Comput. Environ. Urban. Syst.* **24**(3), 232–251 (2000)
- Gandon, F.: Distributed artificial intelligence and knowledge management: Ontologies and multi-agent systems for a corporate semantic web. *Scientific Philosopher Doctorate Thesis in Informatics*, Defended Thursday the 7th of November 2002, INRIA and University of Nice - Sophia Antipolis, Doctoral School of Sciences and Technologies of Information and Communication, (2002)
- Gasevic, D., Djuric, D., Dvedzic, V.: *Model Driven Architecture and Ontology Development*, 328p. Springer-Verlag, Berlin Heidelberg New York (2006).
- Camacho-Hübner, E., Golay, F.: Continuity and evolution of ontologies for urban morphological processes. In: Teller, J., Lee, J., Roussey, C. (eds.) *Ontologies for Urban Development: Interfacing Urban Information Systems*. *Studies in Computational Intelligence*, vol. 61, pp. 143–156. University of Geneva 6,7 Nov 2006. Springer Verlag (2007). ISBN 978-3-540-71975-
- Gómez-Pérez, A., Fernandez-Lopez, M., Corcho, O.: *Ontological Engineering with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. First Edition *Series: Advanced Information and Knowledge Processing* 1st ed. 2004. 2nd printing, 2004, XII, 403 p. 159 illus., Hardcover (2004). ISBN: 978-1-85233-551-9

- Gomez-Pérez, A., Ramos Gargantilla, J.A., Rodríguez Pascual, A., Vilches Blázquez, L.M.: The IGN-E case: Integrating through a hidden ontology Lecture notes in geoinformation and cartography, pp. 417–434 (2008)
- Guizzardi, G., Falbo, R.A., Pereira Filho, J.G.: Using objects and Patterns to implement domain ontologies. *J. Braz. Comput. Soc. (Special Issue on Software Engineering)*, **8**(1) (2002)
- Guizzardi, G., Wagner, G., Guarino, N., van Sinderen M.: “An Ontologically Well-Founded Profile for UML Conceptual Models”, *Lecture Notes in Computer Science*, vol. 3084, (2004)
- IBM: *Ontology Definition Metamodel*, June 2006
- Kaza, N., Hopkins, L.D.: Ontology for land development decisions and plans. In: Teller, J., Lee, J., Roussey, C. (eds.) *Ontologies for Urban Development: Interfacing Urban Information Systems. Studies in Computational Intelligence*, vol. 61, pp. 143–156. University of Geneva, 6,7 Nov 2006. Springer Verlag (2007). ISBN 978-3-540-71975-
- Kim, S.-A.: An ontology based learning system for architectural heritage cases. *J. Archit. Inst. Korea* **21**(10), 97–104 (2005)
- Kim, S.-A., Kim, Y.S.: Design process visualizing and review system with architectural concept design ontology. In: *International Conference on Engineering Design, ICED 2007*, 28–31 Aug 2007. Paris (2007)
- Lacasta, J., Nogueras-Iso, J., Zarazaga-Soria, F.J., Muro-Medrano, P.: Generating an urban domain ontology through the merging of cross-domain lexical ontologies. In: *Proceedings of Second Townontology Workshop “Ontologies for Urban Development: Conceptual Models for Practitioners,”* 17, 18 Oct 2007. Castello del Valentino, Turin, Italy (2007)
- Lammari, N., Comyn-Wattiau, I., Akoka, J.: Extracting generalization hierarchies from relational databases: A reverse engineering approach. *Data. Knowl. Ing.* **63**(2), (2007)
- Lassila, O., McGuinness, D.: *The Role of Frame-Based Representation on the Semantic Web Knowledge Systems Laboratory Report KSL-01-02*, Stanford University, 2001; Also appeared as *Linköping Electronic Articles in Computer and Information Science*, vol. 6, No. 005, Linköping University (2001)
- Lee, J., McMeel, D.: “Pre-ontology” considerations for communication in construction. In: Teller, J., Lee, J., Roussey, C. (eds.) *Ontologies for Urban Development: Interfacing Urban Information Systems. Studies in Computational Intelligence* vol. 61, pp. 143–156. University of Geneva, 6,7 Nov 2006. Springer Verlag (2007). ISBN 978-3-540-71975-
- Li, M., Du, X.-Y., Wang, S.: Learning ontology from relational database. In: *Proceedings of International Conference on Machine Learning and Cybernetics*. vol. 6, pp. 3410–3415, (2005)
- Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A.: *WonderWeb DeliverableD18. The WonderWeb Library of Foundational Ontologies and the DOLCE ontology (final report) (ver. 1.0, 31-12-2003)* <http://www.loa-cnr.it/Publications.html#Pub2003> (2003)
- Mounce, S., Brewster, C., Ashley, R., Hurley, L.: Knowledge management for more sustainable water systems. In: *The Proceedings of the Final Conference of the COST ACTION C21 – TOWNTOLOGY Urban Ontologies for an Improved Communication in Urban Civil Engineering Projects*. March 2009 Liege, pp. 39–49 (2009)
- Nogueras-Iso, J., López, F.J., Lacasta, J., Zarazaga-Soria, F.J., Muro-Medrano, P.R.: Building an Address Gazetteer on top of an Urban Network Ontology. In: Teller, J., Lee, J., Roussey, C. (eds.) *Ontologies for Urban Development: Interfacing Urban Information Systems. Studies in Computational Intelligence*, vol. 61, pp. 143–156. University of Geneva, 6,7 Nov 2006. Springer Verlag (2007). ISBN 978-3-540-71975-
- Object Management Group: *Unified Modeling Language, version 1.5, Mars 2003* – <http://www.omg.org/docs/formal/03-03-01.pdf> (2003)
- Object Management Group: *UML 2.0, OCL specification*, May 2006
- Philippe, M.: *Translations between UML, OWL, KIF and the WebKB-2 Languages (For-Taxonomy, Frame-CG, Formalized English)*, Technical Report, May/June 2003
- Pinet, F., Roussey, C., Brun, T., Vigier, F.: The use of UML as a tool for the formalisation of standards and the design of ontologies in agriculture. In: *Advances in Modeling Agricultural Systems*, 18 p. Springer, New York (2009).

- Pinet, F., Ventadour, P., Brun, T., Papajorgji, P., Roussey, C., Vigier, F.: Using UML for ontology construction: A case study in agriculture. In: The 7th AOS Workshop on Ontology-Based Knowledge Discovery: Using Metadata & Ontologies for Improving Access to Agricultural Information, Bangalore, India, Nov 2006
- Protégé.: Stanford University – <http://protege.stanford.edu> (2005)
- Schreiber, G.: A UML Presentation Syntax for OWL Lite, Technical Report. (2005)
- Sowa, J.: Knowledge Representation: Logical, Philosophical and Computational Foundations. Brooke Cole Publishing Co, Pacific Grove (1999).
- Sure, Y., Staab, S., Studer, R.: On-To-Knowledge methodology. In: Staab S., Studer R. (eds.) Handbook on Ontologies, chapter 6, pp. 117–132. Springer-Verlag, Berlin Heidelberg New York (2003)
- Sure, Y., Studer, R.: OntoKnowledge Project, Deliverable 18: OntoKnowledge Methodology final version, 1999. Available online <http://www.ontoknowledge.org/download/del18.pdf> (1999)
- Trausan-Matu, S.: A socio-cultural ontology for urban development. In: Teller, J., Lee, J., Roussey, C. (eds.) Ontologies for Urban Development: Interfacing Urban Information Systems. Studies in Computational Intelligence, vol. 61, pp. 143–156. University of Geneva 6,7 Nov 2006. Springer Verlag (2007). ISBN 978-3-540-71975-
- Vilches Blázquez, L.M., Bernabé Poveda, M.A., Suárez-Figueroa, M.C., Gómez-Pérez, A., Rodríguez Pascua, A.F.: Towntology & hydrOntology: relationship between urban and hydrographic features in the geographic information domain. In: Teller, J., Lee, J., Roussey, C. (eds.) Ontologies for Urban Development: Interfacing Urban Information Systems. Studies in Computational Intelligence, vol. 61, pp. 143–156. University of Geneva 6,7 November 2006. Springer (2007). ISBN 978-3-540-71975-
- Wildemann, T., Salokhe, G., Keizer, J.: Applying New Trends to the Management of Bibliographic Information on Agriculture - http://www.gil.de/publications/zai/archiv/R8_20040003.pdf (2004)