Chapter 3
Ontologies

The term "ontology" originates in philosophy, where it refers to the study of being or existence and the organization of reality (Guarino and Giaretta 1995; Studer et al. 1998). The term was introduced into engineering through the field of artificial intelligence where it refers to the representation of the real world in computer programs.

Ontology can be understood as a manner of organizing related concepts, as illustrated in Fig. 3.1. Such grouping of concepts has a more specifically defined purpose when compared to general organizations of concepts presented in previous section. The value of ontologies lies especially in the fact that ontologies are intended for organizing concepts under a common specification, often covering a complete domain, in order to facilitate knowledge sharing. The key properties of ontologies are often indicated directly in their definition, which is given in the following section.

3.1 Definition

The term ontology, similarly to the term concept, has been defined from many different viewpoints and with different degrees of formality. Ontologies can be viewed as mediators in the representation of knowledge by means of concepts. Therefore, ontologies lie between concepts (which they subsume) on the one hand and the embracing knowledge domain (within which they are embedded) on the other. Among several proposed definitions that explain ontology in the context of computer science, we prefer the following:

An ontology is a formal and explicit specification of a shared conceptualization.

The definition is coined from two widely adopted definitions provided by Tom Gruber (Gruber 1993) and Willem Borst (Borst 1997). In our opinion, the presented definition is important because it contains the following key aspects of ontologies (Studer et al. 1998):

G. Jakus et al., *Concepts, Ontologies, and Knowledge Representation*,
SpringerBriefs in Computer Science, DOI: 10.1007/978-1-4614-7822-5_3,

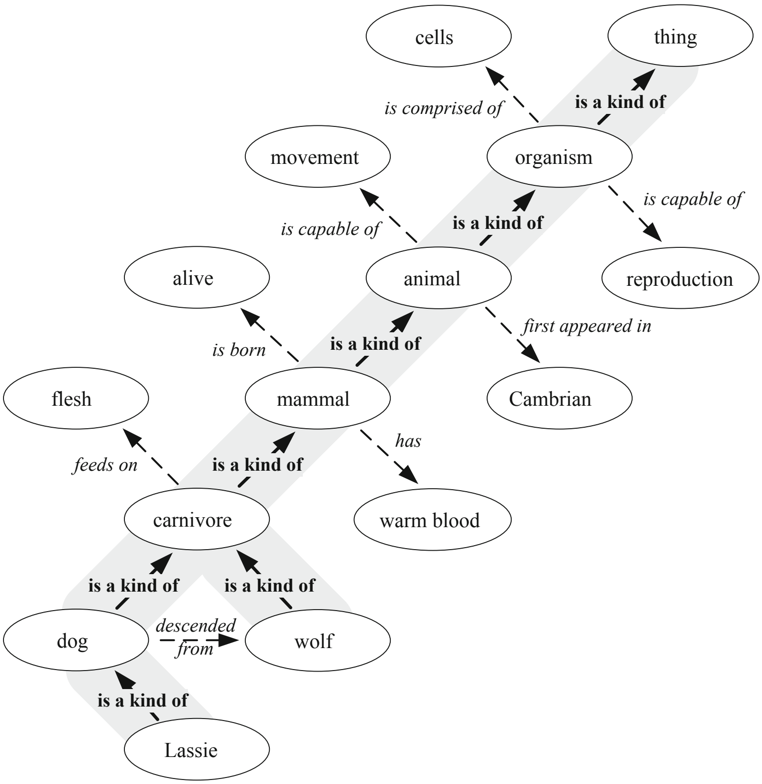

Fig. 3.1 An example of ontology with a highlighted taxonomical structure

- Ontologies are *conceptualizations*. They are abstract models consisting of concepts that are relevant for describing the real world. As such, ontologies act as a sort of surrogate of reality.
- Ontologies are *explicit*, by which we mean that concepts, relations and other components of ontology are defined explicitly.
- Ontologies are *formal* as they are intended to be processed by the computers. The use of natural language is not appropriate due to its ambiguousness, inconsistency and incomplete specification.
- Ontologies are *shared* as they capture consensual knowledge established by a group of interested users. Consequently, by choosing a specific ontology, the user makes a commitment to a set of terms, or *ontological commitments*, which determine how and what to perceive in the reality.

Knowledge sharing is actually one of the key roles of ontologies in computer science. In addition, combined with ontological commitments, the task of knowledge sharing differentiates ontologies from data models which are mostly intended to be used within a single application (Schreiber 2008).

3.2 Organization

Before constructing an ontology, some issues regarding the structure and content of ontologies have to be addressed. Firstly, one needs to choose the proper means to adequately describe a domain by selecting the relevant components of the domain model. Another matter to consider when building an ontology, is the level of its generality and, consequently, the level of its reusability. Finally, the ontology has to be constructed—manually, semi-automatically or even completely automatically. The presented options are the focus of this section.

3.2.1 Ontology Components

The components of ontologies typically include (Studer et al. 1998; Gomez-Perez and Corcho 2002; Navigli et al. 2003; Khoo and Na 2006):

- concepts, classes, collections, sets or types;
- objects, individuals, instances or entities;
- attributes, properties, or features of concepts or objects;
- attribute values;
- relations among classes and/or objects.

Ontology is typically built on top of a *taxonomy*—a hierarchical structure of concepts which limits the relation among the concepts to the formulation "*is a kind of*". Ontology builds on taxonomy by adding a richer network of semantic relations and additional components such as functions, restrictions or constraints, inference rules and axioms. Figure 3.1 shows an example of ontology with its taxonomical framework highlighted. This framework, formed by associating concepts using the relation "*is a kind of*", is further enriched with other semantic relations represented with dashed arrows and italic text.

3.2.2 Types of Ontologies

The definition of ontologies presented in the previous section represents them as a sort of conceptualizations of reality. The definition does not attempt to specify the level of generality of such a conceptualization and neither its scope. To provide an efficient reuse of ontologies and to avoid developing new ontologies when they are already available, it is, nevertheless, useful to divide ontologies in at least two levels of generality.

Generic ontologies (also referred to as *general, upper, foundation, top-level, common* or *core* ontologies) contain knowledge that can be reusable across various domains. The terms and descriptions from the vocabulary of generic ontologies describe very general and domain-independent concepts, such as events, time, space, matter, objects, actions, processes, causality and behavior. Examples of generic ontologies are Cyc (Lenat and Guha 1989; OpenCyc 2012), Dublin Core (DCMI 2012), *Suggested Upper Merged Ontology* (SUMO) (SUMO 2012) and *Descriptive Ontology for Linguistic and Cognitive Engineering* (DOLCE) (Gangemi et al. 2002; LAO 2012).

Specific ontologies represent concepts in a way that is specific to a particular domain, application, task, activity, method, etc. The exact subcategorization within this type of ontologies often depends on a particular field (e.g. knowledge engineering, natural language processing, information retrieval, etc.). In addition, more than one type of specific ontologies can be involved when addressing a particular problem. For example, Guarino (1998) suggested that three distinct specific types of ontology can be involved in building "ontology-driven information systems" (a term by which he referred to all ontology application areas within computer science).

The proposed hierarchical organization of ontologies to be used in ontology-driven information systems consists of domain, task and application ontologies, in addition to the top-level ontology (Fig. 3.2). The vocabularies of domain and task ontologies contain, respectively, the terms representing concepts of a particular domain (e.g. medicine) and a task or an activity (e.g. diagnosing a patient). The concepts in both types of ontologies are specialized from the top-level ontology. Application ontologies, in general, contain subsets of concepts from domain and task ontologies intended to be used in a specific application.

3.2.3 Languages, Tools and Methods for Organizing Ontologies

Ontologies are commonly encoded by using dedicated formal languages which are referred to as *ontology languages*. Ontology languages allow the sharing of knowledge and at the same time support its processing by providing reasoning rules.

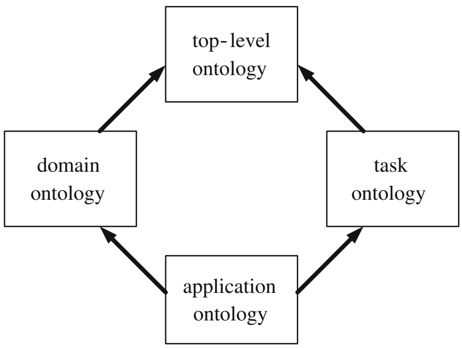

Fig. 3.2 Types of ontologies in an ontology-driven information system (Guarino 1998)

Generally speaking, ontology languages can be classified into two groups depending on the system of logic a particular language is based on. The ontology languages based on *description logic* traditionally originate from the field of artificial intelligence. Two typical examples of such languages are KL-ONE (Brachman and Schmolze 1985) and LOOM (MacGregor and Bates 1987). The other group of ontology languages, based on *first-order logic*, includes CYCL (Lenat and Guha 1989), KIF (Genesereth and Fikes 1992), Ontolingua (Gruber 1993) and Common Logic (CLWG 2012).

In the past fifteen years, a new family of ontology languages was created with the purpose of knowledge sharing on the internet. Examples of these languages include *XML-based Ontology exchange Language* (XOL) (Karp et al. 1999), *Simple HTML Ontology Extension* (SHOE) (Heflin et al. 1999), *Ontology Interchange Language* (OIL) (Fensel et al. 2001) and *Web Ontology Language* (OWL) (OWL 2009).

In addition to using the above-mentioned languages, ontologies can also be represented by conceptual graphs and semantic networks.

Ontologies can be created manually using dedicated software tools or automatically from various existing information resources. Some of the well-known software tools for creating ontologies include TERMINAE (Biebow et al. 1999; TERMINAE 2012), Protégé (Noy et al. 2001; Protégé 2012), KAON (Bozsak et al. 2002; KAON2 2012), DogmaModeler (DM 2012), HOZO (Mizoguchi et al. 2007; HOZO 2012) and OntoStudio (Weiten 2009; Semafora 2012). The methodology for constructing ontologies (also implemented in some of these tools) was developed under the projects OntoClean (Guarino and Welty 2002; OntoClean 2012), On-To-Knowledge (Sure and Studer 2002) and Dogma (Jarrar and Meersman 2002).

Beside the methodology and tools for manual generation of ontologies, several approaches to automatic or semi-automatic ontology generation (also referred to as "ontology learning") from existing resources have also been developed. The resources include:

- text documents (Kietz et al. 2000; Blaschke and Valencia 2002; Fortuna et al. 2007; Navigli et al. 2003);
- web documents (Maedche and Staab 2001; Gal et al. 2004; Navigli and Velardi 2004; Liu et al. 2005); and
- multimedia (Jaimes and Smith 2003).

A more thorough review of the established as well as more recent approaches to ontology construction can be found in the survey of (Barforoush and Rahnama 2012). The present survey, however, continues with examples on the practical use of ontologies.

3.3 Ontology Use

The primary purpose of the effort to define and organize ontologies is to facilitate knowledge sharing. Among numerous cases of use for ontologies, we have chosen to present their application in software agents, natural language processing, media annotation and knowledge extraction.

3.3.1 Software Agents

Software agents are computer programs acting on behalf of human users or other programs. A software agent that has the ability to acquire or apply knowledge to accomplish its goals is referred to as an *intelligent agent*. In this section we give examples of using ontologies to support intelligent agents in pervasive computing environments and Semantic Web services.

3.3.1.1 Intelligent Agents in Pervasive Computing Environments

In pervasive computing environments, agents cooperate with each other and with other devices and services in order to support human activity, goals and needs in an "*anywhere, anytime fashion*" (Chen et al. 2003a). For example, the architecture of the pervasive computing environment proposed in Chen et al. (2003b) is focused around an intelligent agent referred to as Intelligent Context Broker (ICB). The task of an ICB is to establish and share a coherent model of the environment it has been put in, including other devices, agents and people. After the ICB acquires information from sensors, devices and other agents in its environment, it integrates the information from various sources into a coherent model which is eventually used for reasoning and shared with other agents.

The proposed architecture is implemented by using a set of ontologies that provide concepts relevant for modeling pervasive computing environments (Chen et al. 2003a). A part of the ontology supporting the task of an ICB is presented in Fig. 3.3. The ontology is used to model a situation in which a speaker gives an invited speech or a presentation at a meeting which takes place in a particular room in a building. Beside the speaker, the room also hosts other participants representing the audience. The ontology also provides classes that represent environmental agents which are, according to the ontology, responsible for providing environmental information, such as whether a particular room is hosting an event, which specific people are participating in the event and what their intentions are.

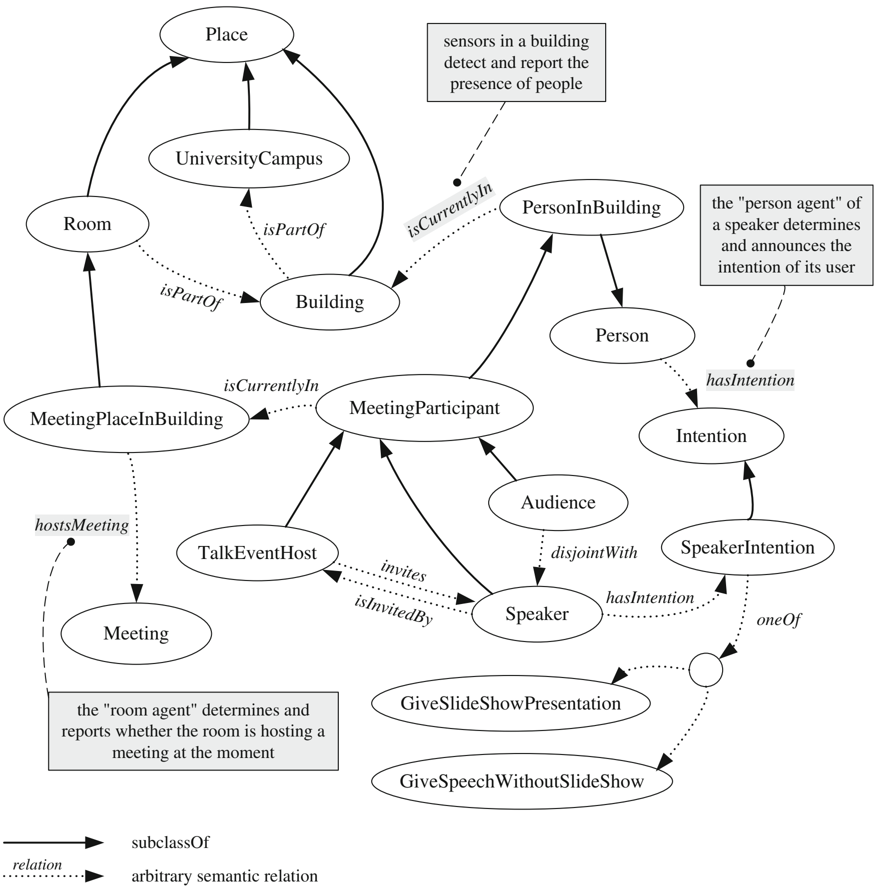

Fig. 3.3 A part of the ontology supporting the intelligent context broker (Chen et al. 2003a)

3.3.1.2 Semantic Web Services

The role of software agents on the Web is to implement web services—i.e. pieces of software that can communicate over the internet and provide functionalities that can be (re)used in web applications. Intelligent software agents and automated services which they provide to web users or other agents are crucial for the implementation of Semantic Web. In order to provide a shared vocabulary supporting the communication among the agents, the *OWL for Services* (OWL-S) ontology was specified (Martin et al. 2004). This ontology uses the OWL language to supplement the existing standards with explicit semantics enabling agents to automatically discover services and their capabilities, to invoke, monitor and compose these services, and to enable their interoperation. The OWL-S ontology is comprised of three lower-level ontologies (Fig. 3.4):

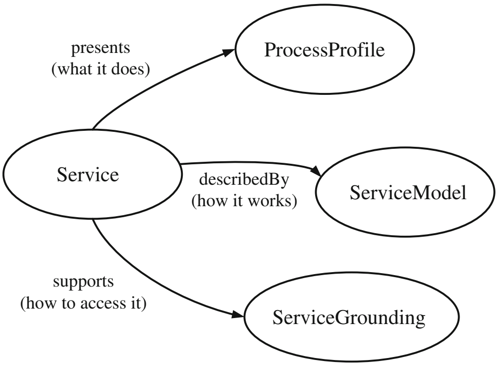

Fig. 3.4 Top level of the OWL-S ontology for describing semantic web services (Martin et al. 2004)

- The *profile ontology* ("ProcessProfile") describes what a service does and it is as such used for advertising and discovering services.
- The *process model ontology* ("ServiceModel") describes how a service is used. It provides the descriptions of how to interact with the service at the abstract level, the outcomes of using a specific service and the conditions under which the desired outcome will occur.
- The *grounding ontology* ("ServiceGrounding") specifies how to interact with a service via messages by providing concrete details such as message format, transport protocol, etc.

The way in which a web service is supposed to be used can be described with no more than one process model. On the other hand, a service can be assigned multiple alternative profiles and groundings, which allows advertising and interacting with the service in different contexts. As an example, Fig. 3.5 presents the most important classes and properties of the profile ontology. According to this ontology, a profile consists of:

- properties that link a profile instance to an instance of the corresponding service;
- human-readable information, such as service name, the contact information of the person responsible and textual description of the service; and
- description of functionality by specifying service inputs, outputs, parameters, preconditions and results, all of which are otherwise defined in the process model ontology.

3.3.2 Natural Language Processing

In the field of natural language processing, a general agreement holds that, whenever dealing with the meaning of texts, ontology should be involved. Ontology is considered as a resource of knowledge about the world (or a domain) consisting of primitive symbols, used for the representation of meaning.

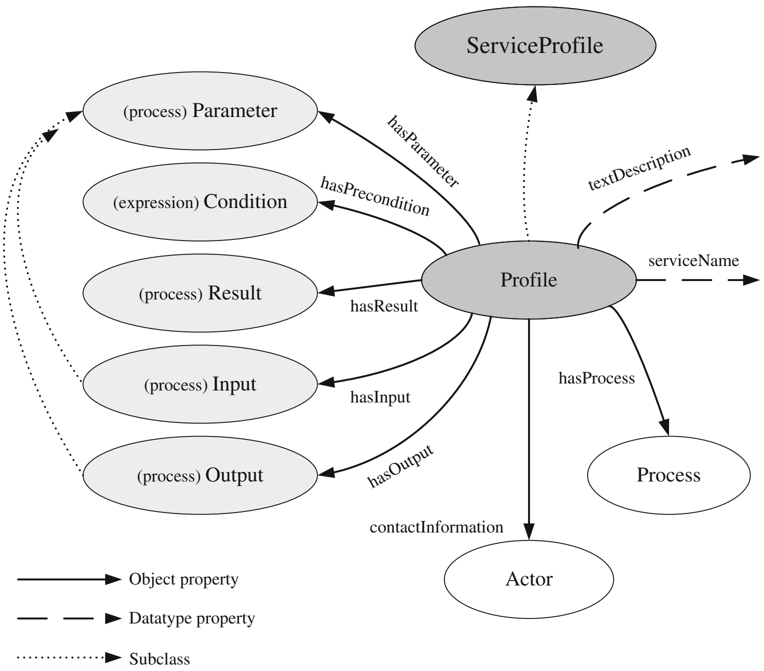

Fig. 3.5 The structure of the top level of the OWL-S profile ontology (Martin et al. 2004)

These symbols stand for concepts which are further interconnected with taxonomic ("*is-a-kind-of*"), semantic and discourse pragmatic relations (Mahesh 1996).

A typical system making use of an ontology is Mikrokosmos (Beale et al. 1995), an analyzer of source texts, which produces a *text meaning representation* (TMR) as the result of the analysis. TMR structures are natural language independent ("interlingual") representations of meaning and can, as such, theoretically be used as the basis for the generation of text in any target natural language.

Figure 3.6 presents the architecture of the Mikrokosmos analyzer and the role of ontology within the system. The analysis is twofold:

- the *syntactic analysis* involves determining syntactic functions and relations among and within particular words in the source text; while
- the *semantic analysis* involves generating TMRs by selecting word meanings which best suit the semantic restrictions from the ontology and lexicon.

The ontology in Mikrokosmos system has the following functions:

- It provides a permanent set of concepts used to represent meaning (Fig. 3.6a). The same set of concepts is expressed in different languages with the help of words from language-specific lexicons.
- It provides the basic building blocks to generate TMR structures (Fig. 3.6b). TMRs are constructed by instantiating the concepts from the ontology, and

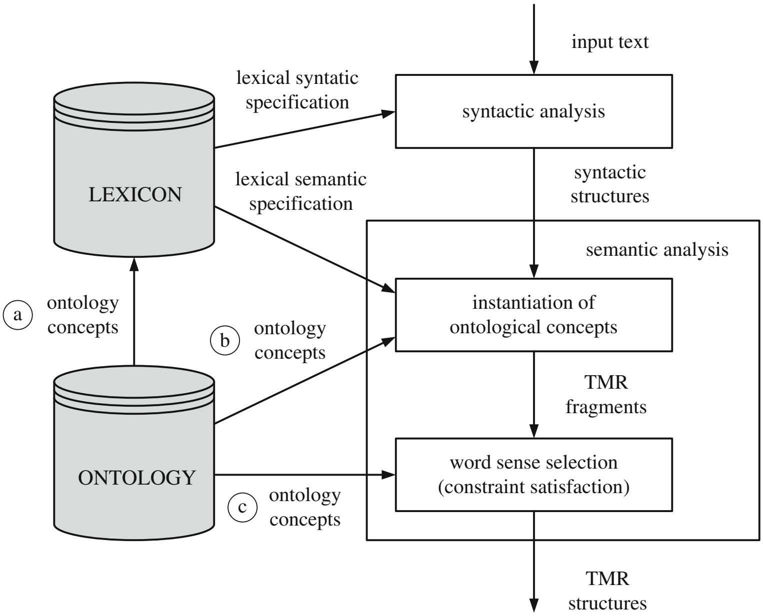

Fig. 3.6 The role of ontology in the architecture of the Mikrokosmos text analyzer (Beale et al. 1995)

supplementing the instances with additional linguistic information, such as aspects, attitudes or modalities.
- It provides constraints for the selection of semantic relations between concepts, which is especially useful for resolving ambiguities (Fig. 3.6c).
- It enables inferences, which can also be used for resolving ambiguities (Fig. 3.6c). In addition, inferences can also be used for filling in the missing textual meaning and for dealing with metonymies, metaphors and complex nominal phrases.

Figure 3.7 presents the taxonomic hierarchy of concepts on the first three levels of the Mikrokosmos ontology.

In the section concerning the methods of constructing ontologies, we mentioned that, among others, ontologies can be constructed automatically from various resources including text documents. Due to the fact that such a method of construction requires techniques of natural language processing supported by ontologies, we present one of the approaches, i.e. OntoLearn.

OntoLearn (Navigli et al. 2003) is a system for ontology learning from textual documents. It consists of the following processes (Fig. 3.8):

- *Extraction of domain terminology*. The language processor first extracts the candidate terms for domain terminology from the domain corpus. The terms are then filtered according to the specificity to a particular domain. The specificity is

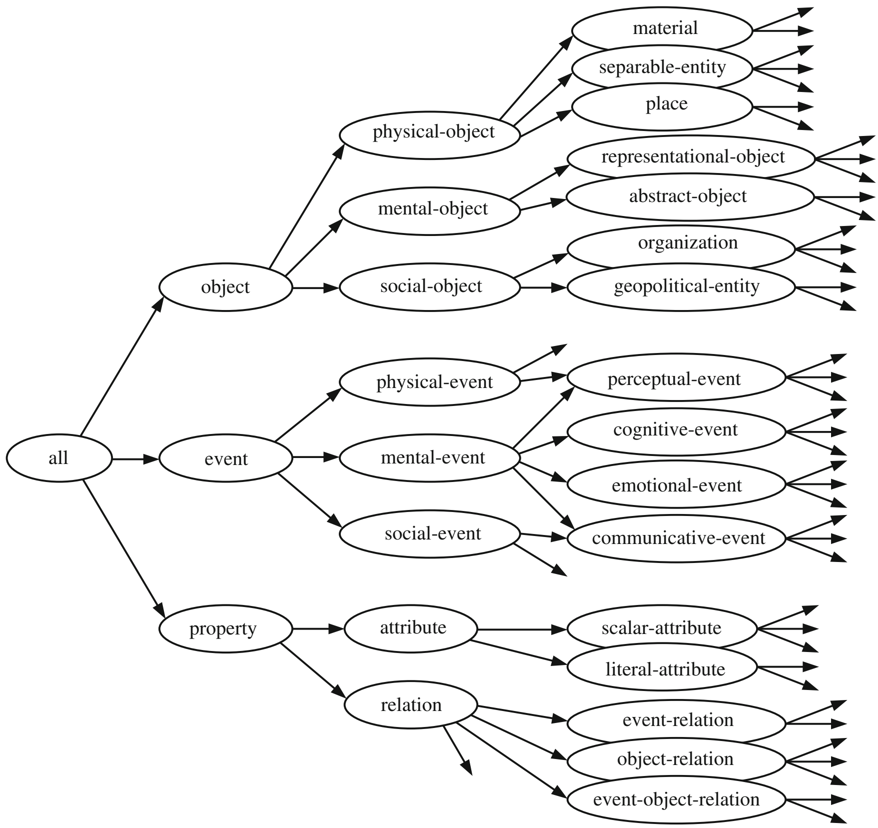

Fig. 3.7 Top-level hierarchy of the Mikrokosmos ontology (Mahesh 1996)

measured in the frequencies of candidate term across different domains, which are calculated with the aid of contrastive corpora.

- *Semantic interpretation*. Semantic interpretation involves finding the sense (concept) behind each component of a complex term. This is achieved by associating each of the term components to the corresponding concept in a generic ontology derived from WordNet.
- *Identification of taxonomic and semantic relations*. The concepts that were recognized in semantic interpretation are interlinked with taxonomic ("*is-a-kind-of*") and other semantic relations, which generates a "*domain concept forest*".
- *Ontology integration*. This process includes creating a specialized domain ontology by extending the generic ontology with the "*domain concept forest*" and removing the concepts that are not considered relevant to the domain.

Besides ontology learning, another practical use of OntoLearn is the translation of multi-word terminology. As the process of semantic interpretation relates individual terms of the compound term to the corresponding concepts (and

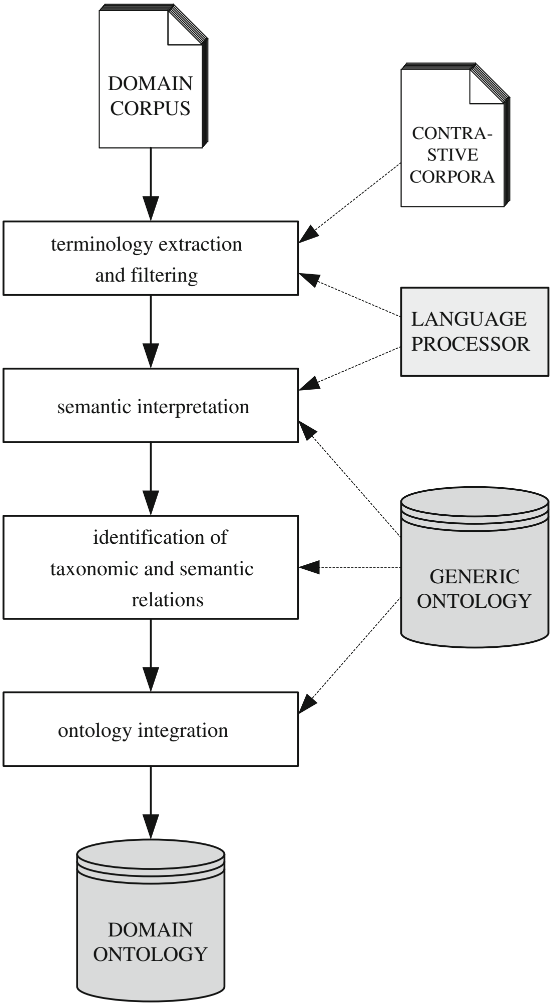

Fig. 3.8 The architecture of OntoLearn (Navigli et al. 2003)

consequently the whole compound term to a compound concept), this resolves the potential ambiguities in the source term. Two terms in different languages that can be matched to the same concepts can be considered mutual translations.

3.3.3 Media Content Annotation

The primary purpose of annotating different types of media, such as text, photos, audio or video, is to describe its content in order to facilitate the content retrieval process. To provide a shared vocabulary for such descriptions, once again, the use ontologies turn out to be a suitable choice. An example of an ontology used for photo annotation is described in Schreiber et al. (2001). The ontology consists of two parts: a photo annotation ontology and subject matter vocabulary.

The photo annotation ontology specifies the organization (template) for photo annotations, which is independent of a particular domain. It consists of the following features:

- A *subject matter feature* is used to describe the subject a photo depicts. The subject matter feature links photo annotation ontology to the subject matter vocabulary.
- A *photograph feature* specifies metadata about the specific circumstances related to the photo, such as how, when and why the photo was taken.
- A *medium feature* represents metadata about the manner in which the photo is stored, including, for example, storage format or resolution.

The subject matter vocabulary acts as a domain-specific ontology used to describe the theme of the photo. It consists of the following four elements:

- an agent (e.g. "an ape"),
- an action (e.g. "eating"),
- an object (e.g. "a banana"), and
- a setting (e.g. "in a forest at dawn").

A more detailed structure of the photo annotation ontology is presented in Fig. 3.9 in the form of a UML class diagram. The white-filled triangle arrow represents the inheritance relation ("*is-kind-of*"). The lines ending with a black diamond represent the composition relations ("*part-of*").

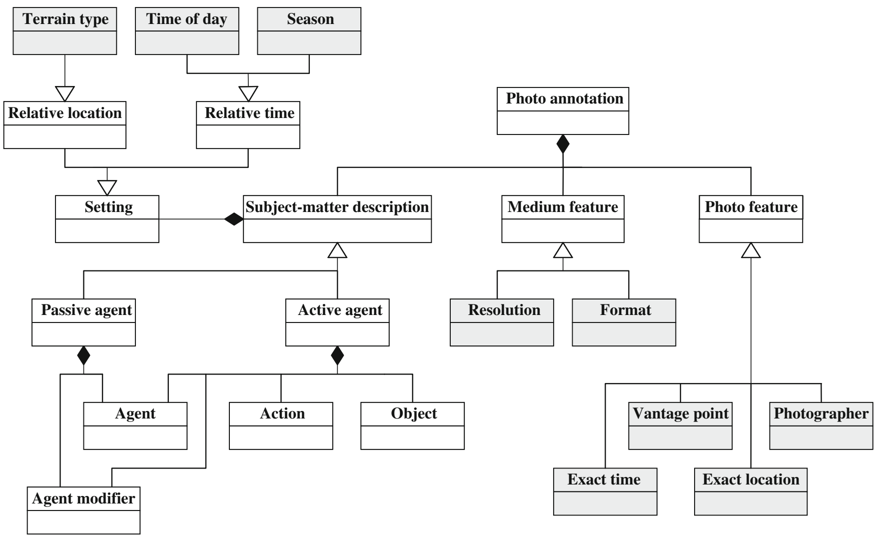

Fig. 3.9 Structure of the photo annotation ontology (Schreiber et al. 2001)

3.3.4 Knowledge Extraction

Knowledge extraction refers to automatically obtaining knowledge from structured or unstructured sources. The obtained knowledge is represented in a formal conceptualized manner according to a formal specification (e.g. an ontology), which enables such knowledge to be used for automated reasoning.

An example of an ontology-based system for extracting knowledge from web pages is Artequakt (Alani et al. 2003). The knowledge extraction process using Artequakt takes place in the following steps (Fig. 3.10a):

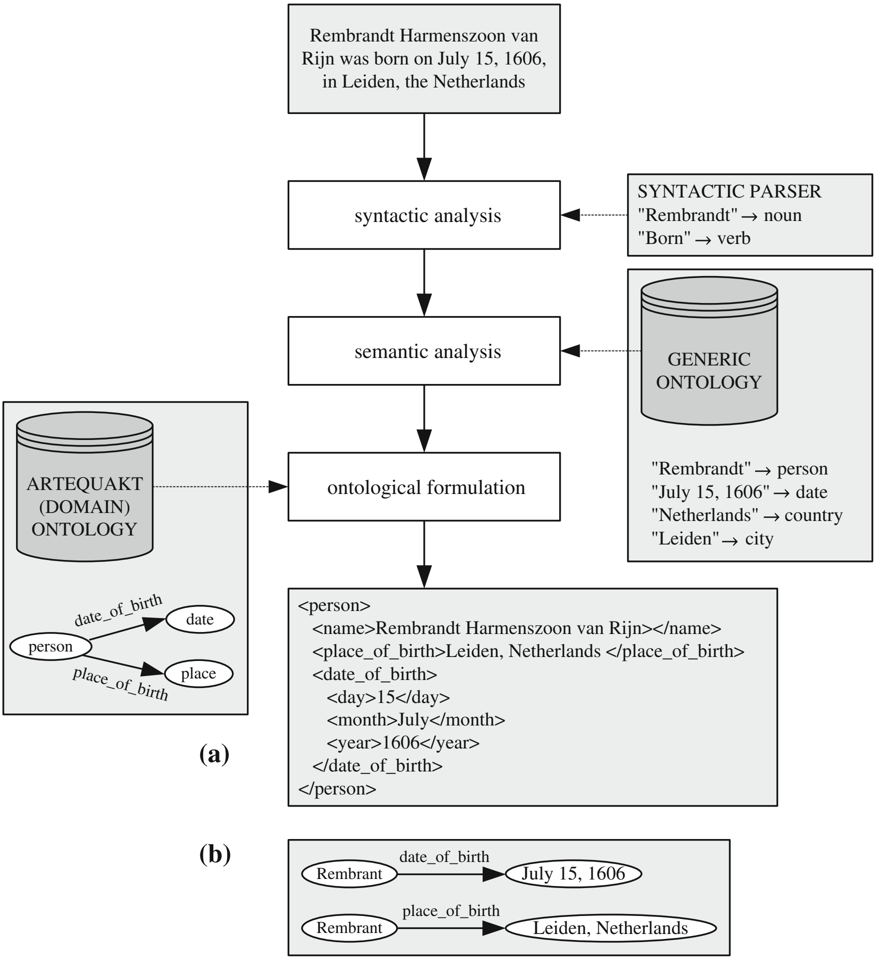

Fig. 3.10 An example of knowledge extraction from a Web page using Artequakt (Alani et al. 2003)

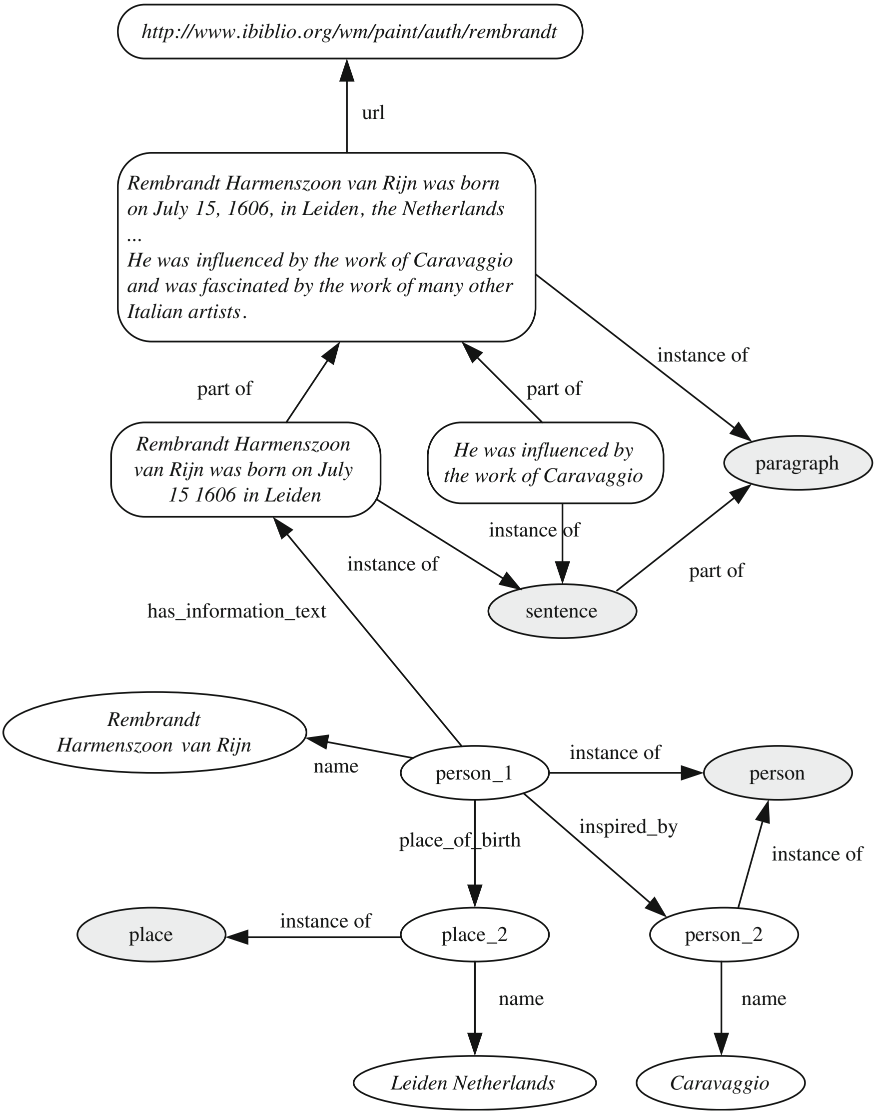

Fig. 3.11 Knowledge base populated with knowledge presented according to the ontology domain representation (Alani et al. 2003)

- In the syntactic analysis, the content of a web page is first divided into paragraphs and sentences, which are then analyzed to determine the grammatical relationships between individual entities.
- Semantic analysis identifies the named entities such as personal and place names, dates, etc. The knowledge needed for such an analysis is obtained from a general-purpose ontology or WordNet.

- In the ontological formulation phase, the semantic relations between pairs of entities are inferred from the domain ontology by matching the entities to the corresponding concepts. The resulting triples consisting of two entities and the associating semantic relation (Fig. 3.10b) are represented in XML.

The knowledge extracted from web pages is presented according to the representation in domain ontology using the relations and the instantiations of the concepts from the ontology. The extracted knowledge represented in such a form is, therefore, suitable for the automatic population of the knowledge base. An example of content from the knowledge base is presented in Fig. 3.11. The gray ovals represent the concepts from ontology, while the white ovals stand for their instantiations (i.e. the entities) representing the knowledge that was extracted from web pages.

This chapter presented various scientific contributions related to ontologies. Ontologies provide a committing set of terms used to represent domain knowledge which is intended to be shared and reused. As such, ontologies can be considered as content theories about the types of things and relations between them that are typical for a specific knowledge domain (Chandrasekaran et al. 1999). On the other hand, however, ontologies do not inherently specify the mechanisms that determine how the represented knowledge should be used in practice. Based on these facts, the next chapter puts ontologies into a broader scope of knowledge representation.

References

Alani H, Kim S, Millard D, Weal M, Hall W, Lewis P, Shadbolt N (2003) Automatic ontology-based knowledge extraction from web documents. IEEE Intell Syst 18(1):14–21

Barforoush AA, Rahnama A (2012) Ontology learning: revisited. J Web Eng 11(4):269–289

Beale S, Nirenburg S, Mahesh K (1995) Semantic analysis in the Mikrokosmos machine translation project. In: Proceedings of the second symposium on natural language processing (SNLP-95), Bangkok, Thailand

Biebow B, Szulman S, Clément A (1999) TERMINAE: a linguistics-based tool for the building of a domain ontology. In: Knowledge acquisition, modeling and management. Lecture notes in computer science, vol 1621. Springer, Berlin, pp 49–66

Blaschke C, Valencia A (2002) Automatic ontology construction from the literature. Genome Inform 13:201–213

Borst WN (1997) Construction of engineering ontologies for knowledge sharing and reuse. Dissertation, Centre for Telematics and Information Technology

Bozsak E, Ehrig M, Handschuh S, Hotho A, Maedche A, Motik B, Oberle D, Schmitz C, Staab S, Stojanovic L, Stojanovic N, Studer R, Stumme G, Sure Y, Tane J, Volz R, Zacharias V (2002) KAON—towards a large scale semantic web. In: E-commerce and web technologies. Lecture notes in computer science, vol 2455. Springer, Berlin, pp 231–248

Brachman RJ, Schmolze J (1985) An overview of the KL-ONE knowledge representation system. Cognitive Sci 9(2):171–216

Chandrasekaran B, Josephson J, Benjamins V (1999) What are ontologies, and why do we need them? IEEE Intell Syst App 14(1):20–26

Chen H, Finin T, Joshi A (2003a) An ontology for context aware pervasive computing environments. Knowl Eng Rev 18(3):197–207
Chen H, Finin T, Joshi A (2003) An intelligent broker for context-aware systems. In: Adjunct proceedings of Ubicomp, Seattle
Common Logic Working Group (2012) Common logic working group documents. http://cl.tamu.edu. Accessed 25 Sep 2012
Dublin Core Metadata Initiative (2012) DCMI Home: Dublin core metadata initiative (DCMI). http://dublincore.org/. Accessed 25 Sep 2012
DogmaModeler (2012) http://www.jarrar.info/Dogmamodeler/. Accessed 25 Sep 2012
Fensel D, Harmelen F, Horrocks I, McGuinness D, Patel-Schneider P (2001) OIL: an ontology infrastructure for the semantic web. IEEE Intell Syst 16(2):38–45
Fortuna B, Grobelnik M, Mladenić D (2007) OntoGen: semi-automatic ontology editor. In: Proceedings of the 2007 conference on human interface, Beijing
Gal A, Modica G, Jamil H (2004) OntoBuilder: fully automatic extraction and consolidation of ontologies from web sources. In: Proceedings of the 20th international conference on data engineering, Boston
Gangemi A, Guarino N, Masolo C, Oltramari A, Schneider L (2002) Sweetening ontologies with DOLCE. In: Knowledge engineering and knowledge management: ontologies and the semantic web. Lecture notes in computer science, vol 2473. Springer, Heidelberg, pp 223–233
Genesereth MR, Fikes RE (1992) Knowledge interchange format, version 3.0, reference manual. Technical report, Computer Science Department, Stanford University
Gomez-Perez A, Corcho O (2002) Ontology languages for the semantic web. IEEE Intell Syst 17(1):54–60
Gruber TR (1993) A translation approach to portable ontology specifications. Knowl Acquis 5(2):199–220
Guarino N (1998) Formal ontology and information systems. In: Proceedings of formal ontology in information systems, Trento, June 1998
Guarino N, Giaretta P (1995) Ontologies and knowledge bases, towards a terminological clarification. In: Mars NJI (ed) Towards very large knowledge bases. IOS Press, Amsterdam, pp 25–32
Guarino N, Welty C (2002) Evaluating ontological decisions with OntoClean. Commun ACM 45(2):61–65
Heflin J, Hendler J, Luke S (1999) SHOE: a knowledge representation language for internet applications. Technical report, University of Maryland
Hozo-Ontology Editor (2012) http://www.hozo.jp/. Accessed 25 Sep 2012
Jaimes A, Smith JR (2003) Semi-automatic, data-driven construction of multimedia ontologies. In: Proceedings of the IEEE international conference on multimedia and expo (ICME), Baltimore, 6–9 July 2003
Jarrar M, Meersman R (2002) Formal ontology engineering in the DOGMA approach. In: Meersman R, Tari Z (eds.) On the move to meaningful internet systems 2002: CoopIS, DOA, and ODBAS. Lecture notes in computer science, vol 2519. Springer, Berlin, pp 1238–1254
KAON2 (2012) KAON2—Ontology management for the semantic web. http://kaon2.semanticweb.org/. Accessed 25 Sep 2012
Karp R, Chaudhri V, Thomere J (1999) XOL: An XML-based ontology exchange language, version 0.4. http://www.ai.sri.com/pkarp/xol/xol.html. Accessed 25 Sep 2012
Khoo C, Na J-C (2006) Semantic relations in information science. Annu Rev Inf Sci Technol 40(1):157–228
Kietz JU, Maedche A, Volz R (2000) A method for semi-automatic ontology acquisition from a corporate intranet. In: Proceedings of the EKAW workshop on ontologies and text, Juan-Les-Pins, Oct 2000
Laboratory for Applied Ontology (2012) DOLCE. http://www.loa.istc.cnr.it/DOLCE.html. Accessed 25 Sep 2012
Lenat DB, Guha RV (1989) Building large knowledge-based systems: representation and inference in the Cyc project. Addison-Wesley Longman, Boston

Liu W, Weichselbraun A, Scharl A, Chang E (2005) Semi-automatic ontology extension using spreading activation. J Univ Knowl Manag 0(1):50–58
MacGregor R, Bates R (1987) The loom knowledge representation language. Technical report ISI/RS-87-188, Information Sciences Institute, University of Southern California
Maedche A, Staab S (2001) Ontology learning for the semantic web. IEEE Intell Syst 16(2):72–79
Mahesh K (1996) Ontology development for machine translation: ideology and methodology. Technical report, Computer Research Laboratory, New Mexico State University
Martin D, Burstein M, Hobbs J, Lassila O, McDermott D, McIlraith S, Narayanan S, Paolucci M, Parsia B, Payne T, Sirin E, Srinivasan N, Sycara K (2004) OWL-S: semantic markup for web services, W3C member submission. http://www.w3.org/Submission/OWL-S/. Accessed 25 Sep 2012
Mizoguchi R, Sunagawa E, Kozaki K, Kitamura Y (2007) A model of roles within an ontology development tool: Hozo. Appl Ontol 2(2):159–179
Navigli R, Velardi P (2004) Learning domain ontologies from document warehouses and dedicated web sites. Comput Linguist 30(2):151–179
Navigli R, Velardi P, Gangemi A (2003) Ontology learning and its application to automated terminology translation. IEEE Intell Syst 18(1):22–31
Noy NF, Sintek M, Decker S, Crubézy M, Fergerson RW (2001) Creating semantic web contents with protege-2000. IEEE Intell Syst 16(2):60–71
OntoClean Central (2012) http://www.ontoclean.org/. Accessed 25 Sep 2012
OpenCyc.org (2012) http://www.opencyc.org/. Accessed 25 Sep 2012
OWL Working Group (2009) OWL 2 web ontology language document overview. W3C recommendation 27 October 2009. http://www.w3.org/TR/owl-overview/. Accessed 25 Sep 2012
The protégé ontology editor and knowledge acquisition system (2012) http://protege.stanford.edu/. Accessed 25 Sep 2012
Schreiber AT, Dubbeldam B, Wielemaker J, Wielinga B (2001) Ontology-based photo annotation. IEEE Intell Syst 16(3):66–74
Schreiber G (2008) Knowledge engineering. In: van Harmelen F, Lifschitz V, Porter B (eds) Handbook of knowledge representation. Elsevier, Amsterdam, pp 929–946
Semafora systems (2012) OntoStudio. http://www.semafora-systems.com/en/products/ontostudio/. Accessed 25 Sep 2012
Studer R, Benjamins VR, Fensel D (1998) Knowledge engineering: principles and methods. Data Knowl Eng 25(1–2):161–197
The Suggested Upper Merged Ontology (SUMO) (2012) Ontology portal. http://www.ontologyportal.org/. Accessed 25 Sep 2012
Sure Y, Studer R (2002) On-to-knowledge methodology. In: Davies J, Fensel D, van Harmelen F (eds) On-to-knowledge: semantic web–enabled knowledge management. Wiley, New York, pp 33–46
TerminaeWiki (2012) http://lipn.univ-paris13.fr/terminae/index.php/Main_Page. Accessed 25 Sep 2012
Weiten M (2009) OntoSTUDIO® as a ontology engineering environment. In: Davies J, Grobelnik M, Mladenic D (eds) Semantic knowledge management: integrating ontology management, knowledge discovery, and human. Springer, Berlin, pp 51–60