13 Ontologies and the Semantic Web

Stephan Grimm[1] · *Andreas Abecker*[1,4] · *Johanna Völker*[2] · *Rudi Studer*[1,3]
[1]FZI Research Center for Information Technology, Karlsruhe, Germany
[2]University of Mannheim, Mannheim, Germany
[3]Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
[4]disy Informationssysteme GmbH, Karlsruhe, Germany

John Domingue, Dieter Fensel & James A. Hendler (eds.), *Handbook of Semantic Web Technologies*,
DOI 10.1007/978-3-540-92913-0_13,

Abstract: Ontologies have become a prominent topic in Computer Science where they serve as explicit conceptual knowledge models that make domain knowledge available to information systems. They play a key role in the vision of the Semantic Web where they provide the semantic vocabulary used to annotate websites in a way meaningful for machine interpretation. As studied in the context of information systems, ontologies borrow from the fields of symbolic knowledge representation in Artificial Intelligence, from formal logic and automated reasoning and from conceptual modeling in Software Engineering, while also building on Web-enabling features and standards.

Although in Computer Science ontologies are a rather new field of study, certain accomplishments can already be reported from the current situation in ontology research. Web-compliant ontology languages based on a thoroughly understood theory of underlying knowledge representation formalisms have been and are being standardized for their widespread use across the Web. Methodological aspects about the engineering of ontologies are being studied, concerning both their manual construction and (semi)automated generation. Initiatives on "linked open data" for collaborative maintenance and evolution of community knowledge based on ontologies emerge, and the first semantic applications of Web-based ontology technology are successfully positioned in areas like semantic search, information integration, or Web community portals.

This chapter will present ontologies as one of the major cornerstones of Semantic Web technology. It will first explain the notion of formal ontologies in Computer Science and will discuss the range of concrete knowledge models usually subsumed under this label. Next, the chapter surveys ontology engineering methods and tools, both for manual ontology construction and for the automated learning of ontologies from text. Finally, different kinds of usage of ontologies are presented and their benefits in various application scenarios illustrated.

13.1 Scientific and Technical Overview: Foundations

Ontologies gained momentum in Computer Science in the recent years, providing a tool for the explicit representation of knowledge that is otherwise implicitly captured in the software that implements an information system. Often, it is beneficial to have such explicit knowledge about the application domain available for the information system to interact with it at runtime and to share it with other software systems. The mechanisms around ontologies allow for such an interaction with and the sharing of explicitly represented domain knowledge.

This section presents the basics of ontologies as used in Computer Science. After clarifying the notion of ontologies, it elaborates on their use for information systems and reviews various types of ontologies.

13.1.1 Notion of Ontologies

Originating from Philosophy, the notion of ontology has found its way into the field of Computer Science, where ontologies are viewed as conceptual yet computational knowledge models. The notion of ontologies as such will be investigated, giving a definition, identifying some essential characteristics and presenting a formal ontology model.

13.1.1.1 Origin and Definition

In this chapter, the uncountable form "ontology" with its origin in philosophy and the countable variant "an ontology" as used in Computer Science, following [1], will be distinguished.

Ontology

In its original meaning in philosophy, *ontology* is a branch of metaphysics and denotes the philosophical investigation of existence. It is concerned with the fundamental question of "what kinds of things are there?" and leads to studying general categories for all things that exist dating back to the times of Aristotle [2].

Transferred to knowledge representation and Computer Science, information systems can benefit from the idea of ontological categorization. When applied to a limited domain of interest in the scope of a concrete application scenario, ontology can be restricted to cover a special subset of the world. Examples of ontological categories in, for example, the technical vehicular domain are "Vehicle," "Car," or "Engine." In this sense, ontology provides a semantic vocabulary to define the meaning of things.

Ontologies

While "ontology" studies what exists in a domain of interest, "an ontology" is a computational artifact that encodes knowledge about this domain in a machine-processable form to make it available to information systems. In various application contexts, and within different communities, ontologies have been explored from different points of view, and there exist several definitions of what an ontology is. Within the Semantic Web community the dominating definition of *an ontology* is the following, based on [3].

Definition 1 (ontology). An ontology *is a formal explicit specification of a shared conceptualization of a domain of interest.*

This definition captures several characteristics of an ontology as a specification of domain knowledge, namely, the aspects of formality, explicitness, consensus, conceptuality, and domain specificity, which require some explanation.

- *Formality* – An ontology is expressed in a knowledge representation language that is based on the grounds of formal semantics. This ensures that the specification of domain knowledge in an ontology is machine-processable and is being interpreted in a well-defined way. The techniques of symbolic knowledge representation, typically built on the principles of logic, help to realize this aspect.
- *Explicitness* – An ontology states knowledge explicitly to make it accessible for machines. Notions that are not explicitly included in the ontology are not part of

the machine-interpretable conceptualization it captures, although humans might take them for granted by common sense.

- *Consensus* – An ontology reflects an agreement on a domain conceptualization among people in a community. The larger the community, the more difficult it is to come to an agreement on sharing the same conceptualization. In this sense, the construction of an ontology is associated with a social process of reaching consensus.
- *Conceptuality* – An ontology specifies knowledge in a conceptual way in terms of conceptual symbols that can be intuitively grasped by humans, as they correspond to the elements in their mental models. (In contrast to this, the weights in a neural network or the probability measures in a Bayesian network would not fit this notion of conceptuality.) Moreover, an ontology describes a conceptualization in general terms and does not only capture a particular state of affairs. Instead of making statements about a specific situation involving particular individuals, an ontology tries to cover as many situations as possible that can potentially occur [4].
- *Domain Specificity* – The specifications in an ontology are limited to knowledge about a particular domain of interest. The narrower the scope of the domain for the ontology, the more an ontology engineer can focus on capturing the details in this domain rather than covering a broad range of related topics.

In summary, an ontology used in an information system is a conceptual yet executable model of an application domain. It is made machine-interpretable by means of knowledge representation techniques and can therefore be used by applications to base decisions on reasoning about domain knowledge.

13.1.1.2 Essential Characteristics of an Ontology

Often, ontologies are visualized and thought of as *semantic networks* that display interrelated conceptual nodes, as exemplarily depicted in ❯ *Fig. 13.1*. There, a fragment of a conceptual model about vehicles and their parts is shown in form of a graph with conceptual nodes and arcs. Intuitively, the network captures knowledge such as "*cars are*

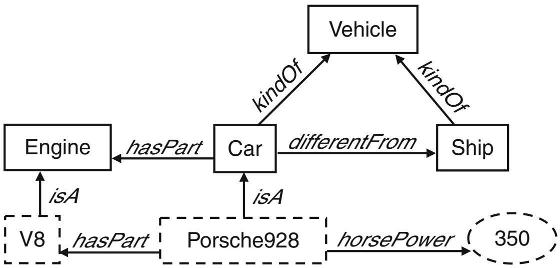

Fig. 13.1
An example ontology as semantic network

kinds of vehicles different from ships and have engines as their parts" or "*a Porsche928 is a car that has a V8 engine and horsepower 350.*"

Although there are various ontology languages with different semantics for similar language constructs next to a multitude of differing ontology formalizations, one can use this semantic network metaphor to identify some *essential characteristics* that are common to most ontologies used in information systems.

- *Interrelation* – Without any interconnection, the plain nodes in a semantic network like the one depicted in *Fig. 13.1* would be rather a loose collection of concepts. *Relations* between concepts allow for such an interconnection, enriching the conceptual nodes with structure. In the example, cars are related to engines by means of a *hasPart* relation.
- *Instantiation* – An essential distinction is the one made between concrete individual objects of the domain of interest and more general categories that group together objects with some common characteristics. *Instantiation* is the mechanism to allow for this distinction by assigning individual objects to classes as their instances, such as the specific Porsche in *Fig. 13.1* is an instance of the class of all cars. This mechanism can be found across almost all forms of conceptual models, sometimes even in the form of meta-modeling allowing for classes to be instances of (meta-) classes themselves.
- *Subsumption* – The most common way of interlinking general conceptual nodes is by *subsumption*, expressing a *kind of* relationship that reflects the notion of specialization/generalization. In the example, cars are stated to be kinds of vehicles, and thus, inherit their properties being their specialization. Subsumption is the mechanism behind the feature of inheritance hierarchies prevalent in conceptual modeling.
- *Exclusion* – A rather sophisticated means of representation is to state "negative" knowledge in form of class exclusion, preventing two general conceptual nodes from overlapping in their extensions. In the example, cars are stated to be different from ships, meaning that being a car excludes being a ship. This form of negative knowledge is found in rather expressive ontology languages that allow for negation.
- *Axiomatization* – The interrelated conceptual nodes as such are often not sufficient to express rich knowledge – they need to take part in complex statements about the domain of interest, which accounts for the notion of *axiomatization*. Besides subsumption, exclusion, or instantiation as simple forms of axiomatization, many ontology languages allow for the formulation of more complex statements in form of general *axioms*. Complex axioms are typically neglected in the graphical presentation of an ontology in lack of an appropriate paradigm for visualization.
- *Attribution* – Although not motivated by the underlying logical formalisms for knowledge representation, the inclusion of statements about datatypes and their values are a common feature in ontology languages, which accounts for the *attribution* of conceptual nodes by strings, numbers, and other data-types. In the example, cars are attributed with a number indicating their horsepower. Datatypes and values are often an indispensable feature in many applications of ontologies.

13.1.1.3 Formal Ontology Model

To present the technical constituents of an ontology in a precise way, a formal ontology model is introduced, defined in [5], that accounts for the essential characteristics identified above. This model is at the same time simple to not present auxiliary characteristics in an overly formal manner, and expressive enough to fit most of the common knowledge representation formalisms and languages used for ontologies in the Semantic Web.

Constituents of an Ontology

An *ontology* $\mathcal{O}$ is a tuple

$$\mathcal{O} = (\mathbf{S}, \mathbf{A}) \tag{13.1}$$

of a *signature* **S** and a set of *axioms* **A**. At this first level of distinction, the elements in the signature, which comprise the conceptual *entities* used for knowledge representation, are separated from the actual statements that use these elements to express knowledge about the domain in the form of axioms. Hence, the entities in an ontology's signature form a semantic vocabulary to be used by knowledge engineers for the formulation of axioms. The axioms themselves are expressed in a specific ontology language or knowledge representation formalism. Often, first-order predicate logic is used as a unifying syntactic framework for expressing axioms.

The signature comprises several sets

$$\mathbf{S} = \mathrm{C} \cup \mathrm{I} \cup \mathrm{P} \tag{13.2}$$

of classes C, instances I and properties P, the distinction of which is an essential feature in almost all forms of conceptual modeling ranging from UML (http://www.uml.org/) in software design to logically expressive ontology languages like OWL (http://www.w3.org/TR/owl2-overview/). Instances map to the individual nodes in a semantic network, representing individual objects of the domain of interest, such as a particular car or person. Classes map to conceptual nodes grouping together instances that have certain characteristics in common, such as the class of all things that are vehicles. Properties map to the customary arcs in semantic networks to express the relation between instances of classes, such as a car having an engine as its part. Hence, at this level of distinction, the notions of instantiation and interrelation are introduced by providing for the respective kinds of vocabulary elements to associate classes with their instances or to interrelate instances via properties. For axioms expressed in first-order predicate logic, classes correspond to unary predicates, properties to binary predicates, and instances to constant symbols, all occurring within the axioms.

The signature entities are further divided into sets

$$\mathrm{C} = \mathcal{C} \cup \mathcal{D}, \quad \mathrm{I} = \mathcal{I} \cup \mathcal{V}, \quad \mathrm{P} = \mathcal{R} \cup \mathcal{T} \tag{13.3}$$

of *concepts* $\mathcal{C}$ and *datatypes* $\mathcal{D}$, *individuals* $\mathcal{I}$, and *data values* $\mathcal{V}$, as well as *relations* $\mathcal{R}$ and *attributes* $\mathcal{T}$. At this third level of distinction, the participants of instantiation, namely, classes and instances, are further split, and abstract concepts are distinguished from datatypes like string or integer, while abstract individual objects of the domain are separated from concrete data values. Accordingly, properties are split into relations, which involve abstract domain objects only, and attributes of classes, which involve also data values. This reflects the natural separation of abstract domain objects from attributes of such objects, which are merely data values attached to them, and thus accounts for attribution. This distinction is also made in entity-relationship modeling in databases, in object-oriented software design, and in ontology languages like OWL, where object properties are separated from datatype properties. In this sense, individuals serve as instances of concepts, while data values serve as instances of datatypes. The explicit distinction between abstract objects of the domain and concrete data values introduced at this level can be found in most forms of conceptual modeling of Computer Science artifacts, and the presence of datatypes is important in many applications.

Specific Axioms

Based on the essential characteristics of an ontology introduced earlier, some special types of axioms are identified that are common to most ontology languages, for which a special notation in *Table 13.1* is introduced. Reference [5] provides mappings for these axiom types to specific ontology languages. Their intuitive meaning is as follows.

- *Instantiation* – An instantiation axiom assigns an instance to a class. The axiom $\alpha_\wedge$ (*Porsche928, Car*), for example, states a particular individual Porsche to be a member of the concept car.
- *Assertion* – An assertion axiom assigns two instances by means of a property. The axiom $\alpha_\rightarrow$ (*Porsche928, horsePower, 350*), for example, asserts the data value 350 to fill a particular Porsche's horsepower attribute.
- *Subsumption*: A subsumption axiom for two classes states that any instance of the subsumed class is also an instance of the subsuming class, while for two properties, it states that any two instances connected by the subsumed property are also connected by the subsuming one. The axiom α_Δ (*Car, Vehicle*), for example, states that any car is also a vehicle.

Table 13.1

Special axioms common in ontology formalisms

Axiom type	Notation	First-order logic expression
Instantiation	$\alpha_\wedge (i, C)$	$[C(i)], i \in \mathrm{I}, C \in \mathrm{C}$
Assertion	$\alpha_\rightarrow (i_1, p, i_2)$	$[p(i_1, i_2)], i_1, i_2 \in \mathrm{I}, p \in \mathrm{P}$
Subsumption	$\alpha_\Delta (E_1, E_2)$	$[\forall x : E_1(x) \rightarrow E_2(x)], E_1, E_2 \in \mathrm{C} \cup \mathrm{P}$
Domain	$\alpha_{D\rightarrow} (p, D)$	$[\forall x, y : p(x, y) \rightarrow D(x)], p \in \mathrm{P}, D \in \mathrm{C}$
Range	$\alpha_{\rightarrow R} (p, R)$	$[\forall x, y : p(x, y) \rightarrow R(y)], p \in \mathrm{P}, R \in \mathrm{C}$
Disjointness	$\alpha_\oplus (C_1, C_2)$	$[\forall x : C_1(x) \wedge C_2(x) \rightarrow \perp], C_1, C_2 \in \mathrm{C}$

- *Domain*: A domain axiom for a property and a class states that for any connection of two instances by that property the source element is an instance of the domain class. The axiom $\alpha_{D\rightarrow}$ (*horsePower*, *Car*), for example, states that the horsepower attribute has domain car.
- *Range*: A range axiom for a property and a class states that for any connection of two instances by that property the target element is an instance of the range class. The axiom $\alpha_{\rightarrow R}$ (*horsePower*, *Integer*), for example, states that the horsepower attribute has range integer.
- *Disjointness*: A disjointness axiom for two classes states that no instance of the one class can also be an instance of the other class, and thus, the classes exclude each other. The axiom $\alpha_{\oplus}$ (*Car*, *Ship*), for example, states the concepts for cars and ships to be disjoint.

Although in general, most ontology languages allow for the formulation of arbitrarily complex and sophisticated axioms to express domain knowledge, this set of basic axioms covers most of what is typically found in the axiomatizations of prevalent Semantic Web ontologies.

13.1.2 Ontologies in Information Systems

Ontologies in information systems build on symbolic knowledge representation techniques known from Artificial Intelligence, and they are used to access domain knowledge provided by means of Web-compliant languages. Next, ontologies will be related to knowledge representation and reasoning and an overview on ontology languages will be given.

13.1.2.1 Ontologies and Formal Knowledge Representation

An ontology – or more precisely its set of axioms – is often seen as a *knowledge base* maintained by a knowledge-based system to have access to and reason about domain knowledge.

Knowledge Representation

Knowledge representation and reasoning aim at designing computer systems that reason about a machine-interpretable representation of the world. Knowledge-based systems have a computational model of some *domain* of interest – their knowledge base *KB* – in which symbols serve as surrogates for real-world domain artifacts, such as physical objects, events, relationships, etc. [6]. The domain of interest can cover any part of the real world or any hypothetical system about which one desires to represent knowledge for computational purposes, while reasoning is performed by manipulating the symbols in the knowledge base. Hence, knowledge representation appears to be an appropriate means for realizing the notion of ontologies in computer systems according to Definition 1.

In contrast to methods of nonsymbolic Artificial Intelligence (like connectionism or statistical machine learning), symbolic knowledge representation builds on processing explicitly modeled pieces of knowledge that are represented in a well-structured way. Therefore, it usually goes along with the manual task of knowledge engineering, and the carefully designed content of a knowledge base is typically of high quality with no noise in data. Alternative approaches that skip manual engineering effort rather rely on a sophisticated runtime interpretation of natural language text or on the automated learning of classifiers from sufficiently large corpora of sample data.

A distinction often made between a knowledge base and an ontology is to see the knowledge base capturing information about a particular state of affairs in the domain, while the ontology captures more general information about any possible situation, as, for example, described in [1]. In this sense, the term ontology is often used for referring to schema knowledge about the classes C and their interrelations P expressed through axioms of type α_{Δ}, $\alpha_{D\rightarrow}$, $\alpha_{\rightarrow R}$ or $\alpha_{\oplus}$, whereas a knowledge base rather comprises plain facts about the concrete instances I expressed through axioms of type $\alpha_{\wedge}$ and $\alpha_{\rightarrow}$. However, various expressive ontology languages blur such a clear distinction by allowing for all kinds of axioms in what they call the specification of an ontology. Therefore, one can see a knowledge base also as a technical means for working with knowledge, in which a knowledge-based system loads (parts of) the specification of an ontology, most likely together with other pieces of knowledge, to take it into account for reasoning. Technically, this amounts to interpreting an ontology's axioms as a knowledge base, that is, $KB = \mathbf{A}$, when reasoning about the domain knowledge it captures.

Reasoning

In knowledge-based systems, the notion of reasoning is associated with the process of reaching conclusions. The axioms that are contained in a knowledge base constitute the *explicit knowledge* a system has about the domain of interest, while the ability to process explicit knowledge computationally by means of reasoning allows the system to derive *implicit* knowledge that logically follows from what has been stated explicitly.

Due to the highly structured form of representation, symbolic approaches to knowledge representation allow for reasoning based on formal logic, which is a powerful tool to simulate the process of reaching conclusions. Logic provides the means to precisely determine what follows from a set of axioms based on formal semantics. Two important aspects of logic-based reasoning with ontologies in the Semantic Web are mainly looked at, namely, the *verification* of an ontology's specification and the *deduction* of new axioms.

- *Verification* – To ensure that an ontology is a good representation of its domain of discourse, reasoning can be used to validate the entirety of axioms in the respective knowledge base at least for their technical soundness. An ontology that contains contradictory information is not considered to be a good domain representation. Reasoning can be used to detect erroneous modeling in an automated way and to report it to knowledge engineers. Erroneous modeling checked for by verification typically comprises logical contradictions or a violation of explicitly stated constraints.

- *Deduction* – Based on the assumption that an ontology correctly represents the domain of interest, the process of deduction derives implicit conclusions that hold in any situation coherent with the axioms in the respective knowledge base, capturing the notion of *logical consequence*. Reasoning constitutes the primary mechanism to determine the deductive conclusions from the symbols and structure of axioms in a knowledge base. The form of statements derived as deductive consequences can thereby range from simple facts to complex generic axioms. An axiom α following from a knowledge base *KB* is typically denoted by $KB \models \alpha$ using the entailment symbol.

The basic operations a knowledge-based system can perform on its knowledge base are usually called `tell` and `ask` [7]. The `tell`-command adds a new statement to the knowledge base, whereas the `ask`-command is used to query what is known. While the `tell`-command determines the explicit knowledge, the `ask`-command operates under deductive closure; that is, it also gives answers that constitute implicit knowledge found by reasoning. In this sense, the command `tell` (KB, α) performs the operation $KB := KB \cup \alpha$ on a knowledge base *KB* and an axiom α, while the command `ask`(KB, α) yields true if and only if α is entailed by *KB*, i.e., $KB \models \alpha$.

13.1.2.2 Usage of Ontologies

Ontologies are used by information systems to be queried for domain knowledge. Although they seem to be very similar to other conceptual models, their ability to access implicit knowledge renders them as distinct artifacts used for building intelligent computer systems.

Ontologies Versus Other Conceptual Models

At a first glance, ontologies seem to be very similar to other kinds of conceptual models used in Computer Science, especially when looking at the typical constructs of established ontology languages. Notions such as interrelation, instantiation, or subsumption can also be found in, for example, UML class diagrams for technical specifications of information systems or in entity-relationship diagrams for the specification of database schemas. However, there is a subtle difference between ontologies and these other forms of conceptual models that is primarily motivated in terms of usage and purpose. Namely, conceptual models known from software and database engineering are prescriptive in that they are means to construct a technical system anew, whereas ontologies are descriptive capturing the knowledge observed to characterize a domain.

The main purpose and goal of designing software systems using UML class diagrams is to prescribe the technical components of an information system that is to be run on a (physical) computer system. Once the system runs, the conceptual model behind it has fulfilled its function and is not consulted or modified at runtime.

A similar argument holds true for conceptual models in database management systems: Entity-relationship diagrams as a conceptual model are often used only at *system-design time* as a preliminary step before designing the relational schema as a basis for efficient storage and data access. They are not intended to be used or changed when the information system is in use.

In contrast to these other conceptual models, the primary purpose of an ontology is to serve as a source of domain knowledge to be queried by an information system that bases *runtime* decisions on the respective answers. By means of performing reasoning, these answers include implicit as well as explicit knowledge. This difference typically imposes additional requirements on the expressivity of knowledge representation formalisms and ontology languages, since the task of representing domain knowledge as such is very general and not restricted to specific aspects of processing, such as imperative runtime behavior or efficient data access. More expressive modeling constructs like negation or class exclusion are, therefore, typically not found outside ontology engineering.

Reasoning with Ontologies

An information system interacts with an ontology $\mathcal{O} = (\mathbf{S}, \mathbf{A})$ via the `tell`-/`ask`-interface. Whenever it encounters new knowledge, it applies the `tell`-command to the set of axioms **A**, which might also introduce new symbols in the signature **S**. When querying for knowledge, it makes use of the `ask`-command, which answers under deductive closure of **A** seen as a knowledge base. For asking, the two reasoning tasks of verification and deduction apply.

Verification – Applied to ontologies, the reasoning task of verification checks whether the set of axioms **A** forms a logically sound specification free of contradictory information and over-constrained restrictions on an ontology's entities. As an example, consider the ontology given in Table 13.2. Here, the restrictions on the individual *SportiveMinibus* in $\mathbf{S}_1$ are over-constrained by the axioms in $\mathbf{A}_1$, yielding a contradiction: On the one hand, the sportive minibus is stated to be a sports car; on the other hand, it is also stated to be van, while all vans are family cars; hence, the sportive minibus is also a family car, and since family cars are disjoint from sports cars, this leads to a logically inconsistent situation.

Deduction – The reasoning task of deduction allows for drawing conclusions from an ontology's specification, thus providing access to the implicit knowledge captured in its axioms. As an example, consider the ontology given in Table 13.3. Here, various

Table 13.2

Example of a contradictory ontology

Ontology $\mathcal{O}_1 = (\mathbf{S}_1, \mathbf{A}_1)$	
Signature $\mathbf{S}_1$	
$\mathcal{C}_1$ = {*Van*, *FamilyCar*, *SportsCar*}, $\mathcal{I}_1$ = {*SportiveMiniBus*}	
Axioms $\mathbf{A}_1$	
α_Δ (*Van*, *FamilyCar*)	Vans are family cars
$\alpha_\oplus$ (*FamilyCar*, *SportsCar*)	Family cars and sports cars are different things
$\alpha_\wedge$ (*SportiveMinibus*, *Van*)	The sportive minibus is a van
$\alpha_\wedge$ (*SportiveMinibus*, *SportsCar*)	The sportive minibus is a sports car

Table 13.3

An example ontology for deduction

Ontology $\mathcal{O}_2 = (\mathbf{S}_2, \mathbf{A}_2)$	
Signature $\mathbf{S}_2$	
$\mathcal{C}_2$ = {*SportsCar*, *Car*, *Vehicle*, *Engine*}, $\mathcal{I}_2$ = {*Porsche928*, *V8*}, $\mathcal{R}_2$ = {*hasEngine*}	
Axioms $\mathbf{A}_2$	
α_{Δ} (*SportsCar*, *Car*)	Sports cars are cars
α_{Δ} (*Car*, *Vehicle*)	Cars are vehicles
$\alpha_{\rightarrow R}$ (*hasEngine*, *Engine*)	hasEngine ranges over engines
$\alpha_{\wedge}$ (*Porsche928*, *SportsCar*)	The Porsche928 is a sports car
$\alpha_{\rightarrow}$ (*Porsche928*, *hasEngine*, *V8*)	The Porsche928 has a V8 engine

conclusions can be drawn from the explicitly stated axioms when, for example, a first-order logic semantics is assumed. Since the individual *Porsche928* is stated to be a sports car and sports cars are stated to be special kinds of cars, it can be concluded to also be a car, that is, $\mathbf{A}_2 \vDash \alpha_{\wedge}$ (*Porsche928*, *Car*). And since cars are stated to be special kinds of vehicles, it can furthermore be concluded to be a vehicle as well, that is, $\mathbf{A}_2 \vDash \alpha_{\wedge}$ (*Porsche928*, *Vehicle*). As for conclusions at class level, it can be inferred that any sports car is also a vehicle, that is, $\mathbf{A}_2 \vDash \alpha_{\Delta}$ (*SportsCar*, *Vehicle*). Moreover, since the range of the property *hasEngine* is stated to be the class *Engine*, the V8 engine, which is assigned to be the engine of a particular car, can be concluded to be an engine, that is, $\mathbf{A}_2 \vDash \alpha_{\wedge}$ (*V8*, *Engine*). In summary, querying an ontology under the mechanism of deduction yields more knowledge than that explicitly stated due to reasoning.

13.1.2.3 Ontology Languages

In the recent years, a landscape of specialized languages for expressing ontologies has evolved, mostly in the context of Semantic Web research. Besides covering the essential characteristics of an ontology as a knowledge representation artifact, they also account for Web-enabling features such as a unique identification of entities via URIs or XML serialization formats. The most prevalent of these Semantic Web ontology languages will be surveyed briefly.

RDF(S) – As an effort for the standardization of metadata, the Resource Description Framework **RDF** [8] and a simple ontology language **RDFS** (RDF Schema) [9] emerged early in the context of the Semantic Web as an initiative from the World Wide Web Consortium (W3C, http://www.w3.org/). Meanwhile, **RDF(S)** has become a well-established and widely accepted standard for encoding metadata and basic ontologies on the Web (see also Semantic Annotation and Retrieval: RDF).

As an ontology language, RDFS has the expressivity for the features of interrelation, instantiation, and subsumption, by hierarchies that can be built over resource classes and properties. It also allows for attribution in reference to XSD datatypes but restricts axiomatization to domain and range restrictions besides subclassing and typing. In particular, RDFS does not exhibit the feature of expressing exclusion or negation of any form, which renders it as a semantically rather lightweight formalism.

OWL – On top of RDF(S), W3C standardization efforts have produced the Web Ontology Language (**OWL**) [10] as the currently most prominent language to express ontologies for their use in the Web. OWL comes in several variants with increasing expressiveness, of which only the most expressive one, namely **OWL Full**, has a proper layering on top of RDF(S), allowing for features of meta-modeling and reification, while the variant **OWL DL** is focused on the formalism of description logic (DL) [11]. In its newest version **OWL 2**, the Web Ontology Language has recently undergone the final steps of W3C standardization.

Besides the class membership and subsumption relations inherited from RDF(S), OWL offers the construction of complex classes from simpler ones by means of logical expressions, which accounts for rich axiomatization including class exclusion. The design goals behind OWL are primarily driven by expressive description logics as decidable fragments of first-order predicate logic (see also KR and Reasoning on the Semantic Web: OWL).

Rule Languages – Besides ontologies as characterized by the W3C standard languages, there is the logic programming paradigm of knowledge representation based on *rules* with an if-then reading, which accounts for a special form of axiomatization. Rule-based systems and deductive databases have brought forth specific languages for expressing rules, such as **F-Logic** (Frame Logic) [12] or **Datalog** [13]; however, these are not tailored to Web standards. An attempt to integrate rules with Web ontologies is the Semantic Web Rule Language (**SWRL**) as a W3C member submission to build rules into OWL ontologies. Suggestions for a restricted use of SWRL-style rules on top of OWL are, for example, DL-Safe Rules [14] or DL Rules [15]. Rule languages for the Semantic Web are further discussed in W3C's Rule Interchange Format (RIF) Working Group with the goal to establish **RIF** as a Web standard for rule languages complementing RDF(S) and OWL (see also KR and Reasoning on the Semantic Web: RIF).

WSML – The Web Service Modeling Language (**WSML**) [16] is an attempt to standardize ontology languages for the Web with a special focus on annotating Semantic Web Services [17]. Next to the service-specific aspects, it provides expressive means for the formulation of ontologies in general, addressing various knowledge representation paradigms in different language variants (see also Semantic Web Services).

Semantic Vocabularies – Apart from expressive knowledge representation formalisms, there are also languages to form controlled semantic vocabularies, which can be used to formulate specific lightweight ontologies with rather few but easy-to-use semantic features. One such effort is the Simple Knowledge Organization System (**SKOS**) [18], which is built on top of the RDF-based W3C standards. Another lightweight approach for expressing semantic networks on the Web is the ISO-standard **Topic Maps** with a focus on a graph-oriented paradigm of representing and visualizing knowledge.

13.1.3 Variants of Ontologies

Since ontologies in Computer Science are a rather new phenomenon, the notion of an ontology is perceived in a broad sense. Many different forms of conceptual models are arguably interpreted as ontologies, varying in different dimensions. Varying forms of ontology appearance, scope, and degree of formality will be discussed next.

13.1.3.1 Varying Form of Appearance

When engineered for or processed by information systems, ontologies appear in different forms. A knowledge engineer views an ontology by means of some graphical or formal visualization, while for storage or transfer, it is encoded in an ontology language with some machine-processable serialization format. A reasoner, in turn, interprets an ontology as a set of axioms that constitute a logical theory.

Graphical Appearance

In knowledge engineering tools, an ontology is often visualized as some form of semantic network with interlinked conceptual nodes. *Figure 13.1* gives an impression of such a graphical visualization. Most typical for such a visualization is to display the taxonomic hierarchy of domain concepts and the customary relations between them. However, certain information, such as the knowledge captured in complex axioms, cannot easily be displayed graphically because of the lack of appropriate visualization paradigms.

Formal Appearance

As ontology languages like OWL are based on logical formalisms, the formal semantics of the language precisely defines the meaning of an ontology in terms of logic. To a reasoner, therefore, an ontology appears as a set of logical formulas that express the axioms of a logical theory. The following logical formulas constitute a set of axioms that formalize the knowledge from the semantic network in *Fig. 13.1* in the description logic formalism that underlies, for example, the language OWL.

$$\begin{gathered}
\ldots \\
Car \sqsubseteq Vehicle \\
Ship \sqsubseteq Vehicle \\
Car \sqcap Ship \sqsubseteq \bot \\
Car \sqsubseteq \forall\ hasPart.Engine \\
\exists\ hasEngine.\top \sqsubseteq Car \\
Car\,(Porsche928),\ Engine\,(V8) \\
hasEngine\,(Porsche928,\ V8),\ horsePower\,(Porsche928,\ 350) \\
\ldots
\end{gathered}$$

This form of appearance of an ontology is free of syntactical or graphical additions or ambiguities and reflects the pure knowledge representation aspect.

Machine-Processable Appearance

When exported for storage on disk or for transfer over the wire, an ontology's specification needs to be expressed in a machine-processable representation format. Hence, to the developer of an ontology editor, storage facility or reasoning tool, an ontology appears in form of some serialization format suitable for machine processing. The following listing displays the information from the formal view above in the RDF XML serialization format.

```
...
<owl:Class rdf:about="#Vehicle"/>
<owl:Class rdf:about="#Car">
    <rdfs:subClassOf rdf:resource="#Vehicle"/>
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasEngine"/>
            <owl:allValuesFrom rdf:resource="#Engine"/>
        </owl:Restriction>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="#Ship"/>
</owl:Class>
<owl:Class rdf:about="#Ship">
    <rdfs:subClassOf rdf:resource="#Vehicle"/>
</owl:Class>
<owl:Class rdf:about="#Engine"/>
<owl:ObjectProperty rdf:about="#hasEngine">
    <rdfs:domain rdf:resource="#Car"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="#horsePower"/>
<Car rdf:about="#Porsche928">
    <horsePower>350</horsePower>
    <hasEngine rdf:resource="#V8"/>
</Car>
<rdf:Description rdf:about="#V8"/>
...
```

It shows the various interconnected concepts, properties, and individuals in their technical XML rendering to be parsed by ontology tools that can read OWL in its RDF XML serialization.

13.1.3.2 Varying Scope

As the domain of an ontology can cover any topic that suits conceptual modeling, there is no restriction to what kind of knowledge can be represented. However, ontologies have been classified in different types according to the kind of knowledge they capture.

The most established distinction is between so-called *upper-level ontologies*, which describe very general categories that can be used across domains, and *domain ontologies*, which capture the knowledge of a specific domain.

Upper-Level Ontologies

Upper-level ontologies attempt to describe very abstract and general concepts that can be shared across many domains and applications. Upper-level ontologies are sometimes also called top-level ontologies or foundational ontologies. They borrow from philosophical notions, describing top-level concepts for all things that exist, such as "physical object" or "information object," as well as generic notions of common-sense knowledge about phenomena like time, space, processes, etc. They are usually well thought out and extensively axiomatized.

Due to their generality, upper-level ontologies are typically not directly used for conceptual modeling in applications but reused as a basis for building more specific ontologies. For a concrete application, it would in most cases not make sense to directly use an upper-level ontology as it is free of any domain knowledge. Often, upper-level ontologies are employed to merely guide the design of domain ontologies, prescribing various knowledge representation patterns that can be instantiated.

Domain Ontologies

Domain ontologies capture the knowledge within a specific domain of discourse, such as medicine or geography. In this sense, they have a much narrower and more specific scope than upper-level ontologies. Prominent ontologies exist, for example, in natural sciences, such as medicine, genetics, geographic and communal efforts such as environment information, tourism, as well as cultural heritage and museum exhibits, to name just a few.

Domain ontologies can be used together with upper-level ontologies in a combined way. In such a case, the domain ontology is typically aligned with the upper-level ontology, meaning that its specific domain concepts subclass or instantiate broader top-level notions inheriting their upper-level axiomatization.

13.1.3.3 Varying Degree of Formality

Ontologies used in the context of the Semantic Web can also be distinguished according to their degree of formality. An ontology's degree of formality determines to which extent it is axiomatized by means of logical statements about the domain. *Lightweight* ontologies possess no or only a few axioms constraining the use of the entities in their signature. On the other hand, *heavyweight* ontologies are characterized by extensive axiomatization for interrelating the signature elements in a sophisticated way – such that nearly all entities are accompanied by many axioms constraining their use and supporting reasoning about them.

Thesauri

The least formal form of an ontology are Thesauri, which organize the words used in a certain domain according to lexical criteria. Examples are language-specific dictionaries that also encode information about synonyms to be used in word-processing software, or a classification of medical terms for diseases. The expressivity of Thesauri is rather low in terms of logic-based knowledge representation and is typically restricted to lexical relations between words, such as synonymity or homonymity. One of the most well-known and comprehensive general-purpose thesauri is **WordNet** [19].

Concept Schemes

As informal semantic networks of interlinked conceptual nodes, concept schemes often evolve as the result of collaborative tagging activities in a Web context. Examples are tag taxonomies [20] and informal hierarchies, for example, modeled in SKOS. They are of low expressivity, offering rather informal semantic relations and means for classification with very limited possibilities for axiomatization. This makes them well-suited to rather uncontrolled environments of collaborate editing and maintenance within a larger community of uncoordinated knowledge contributors.

Taxonomies

Taxonomies are often used for a formalized hierarchical organization of domain knowledge by means of class hierarchies based on the notion of subsumption. An example is a catalog of product categories that builds up a strict subsumption hierarchy of product classes. The main feature of taxonomies is their strict hierarchical categorization of classes. Therefore, the subsumption relationship is typically formalized logically, for example, in terms of transitivity, while no or only few other cross-relations between classes are allowed.

Conceptual Data Models

In Computer Science, the use of various conceptual models tailored toward designing information systems or database management systems is ubiquitous. Examples are entity-relationship diagrams or UML diagrams used for domain modeling. Such conceptual models typically have the expressivity for structuring a domain for the data used within a software system by means of concept subsumption hierarchies and properties and attributes of domain classes. If any logical formalization occurs, it is typically used for checking constraints on the conceptual model to identify faulty data situations rather than for drawing conclusions out of explicit knowledge.

Rule and Fact Bases

In many applications, rule or fact bases serve as data-intensive knowledge bases built for the handling of large numbers of individuals with some basic reasoning. Examples are logic programming rule bases for the derivation of instantiation and assertion axioms, or description logic A-Boxes and RDF(S) graphs for querying facts with simple reasoning over class and property hierarchies. These ontologies typically have the expressivity for interrelating and typing instances and for a rule-based derivation of facts by means of logic programming mechanisms.

General Logical Theories

The most formal and expressive type of an ontology is a general logical theory, in which the domain of discourse is highly axiomatized in an expressive logic-based knowledge representation formalism, such as first-order predicate logic or even higher-order or modal logics. An example is the formal specification of an upper-level ontology with a rich axiomatization for very general notions in the form of modal logic axioms. A general logical theory is not restricted to the derivation of facts at the instance level, but also captures a rich axiomatization about classes and properties at the schema level, allowing for drawing conclusions about general situations in the domain in the form of complex axioms.

13.1.3.4 Some Example Semantic Web Ontologies

In different areas of the Semantic Web particular ontologies have evolved fitting in different categories of the ones reported above.

Upper-Level Ontologies

As for upper-level ontologies, the **Descriptive Ontology for Linguistic and Cognitive Engineering** (**DOLCE**) [21] is the most prominent approach based on philosophical notions and with a representation in OWL. Another established top-level ontology is the **Suggested Upper Merged Ontology** (**SUMO**) [22] with a special link to the WordNet lexicon.

Catalog Schemes and eBusiness Ontologies

For the unifying categorization of materials, products, and services across enterprises, particular coding scheme standards have been established. Two such efforts are the **United Nations Standard Products and Services Code** (**UNSPC**) for use across enterprises throughout the global eCommerce marketplace focused on the American economy, and the similar but more expressive **eCl@ss** standard. There have been efforts to express these product categorization schemes in ontology languages like OWL, as reported in [23]. Based on this, the **GoodRelations** lightweight ontology can be used for annotating eBusiness offerings on the Web, as a non-toy vocabulary for describing the types of goods as well as terms and conditions for buying items and services offered on the Web [24].

Medical Domain Ontologies

The medical domain is one for which many ontologies are available for use in a Web context. **GALEN** is an ontology for clinical information whose encoding in OWL is widely used. The **Systematized Nomenclature of Medicine** (**SNOMED**) is a computer processable collection of medical terminology covering most areas of clinical information such as diseases, findings, procedures, microorganisms, pharmaceuticals, etc., expressed in the description logic formalism. Also the **Foundational Model of Anatomy** (**FMA**) is a medical ontology concerned with the representation of classes or types and relationships necessary for the symbolic representation of the phenotypic structure of the human body. The **International Classification of Diseases** (**ICD**) is a taxonomy of medical terms of diseases for use in diagnostic classification, coming with an OWL representation. The **Unified Medical Language System** (**UMLS**) is a compendium of many controlled vocabularies in the biomedical area. The *UMLS Meta-thesaurus* comprises over one million biomedical concepts and five million concept names, all of which stem from one of the more than 100 incorporated controlled vocabularies and classification systems (including SNOMED, ICD, MeSH – the Medical Subject Headings), the Gene Ontology, and many more. It provides a mapping structure among the source vocabularies and thus allows one to translate among the various terminology systems; the *UMLS Semantic Network* adds a basic semantic structure by associating to each concept one out of 135 so-called *semantic types* (like, e.g., organisms, anatomical structures, biologic function, chemicals, etc.) and also incorporates additional semantic relationships. Finally, the *UMLS SPECIALIST Lexicon* further provides facilities for natural language processing by adding further lexical information.

Service Description Ontologies

For the semantic annotation of Web Services, various service-specific ontologies have been proposed. One is **OWL-S** as an effort to define an ontology for Semantic Web Services markup expressed in OWL. Another one is the **Web Service Modeling Ontology** (**WSMO**) [25], which is expressed in its own specific language WSML. **SAWSDL** is a W3C recommendation which defines a set of extension attributes for the Web Services Description Language (WSDL) and XML Schema definition language that allows one to describe additional semantics of WSDL components. The specification defines how semantic annotation is accomplished using references to external semantic models, for example, ontologies.

Social Web Ontologies

There is a couple of relatively simple, but very widespread RDF schemas that aim at semantically enriching information in the Social Web, making social Web information better interoperable and interpretable, thus better connecting people in the Social Web. Most importantly, **FOAF** (Friend-of-a-fried ontology) is an RDF Schema describing people, their activities, and their relations to other people and objects. Anyone can use FOAF to describe themselves. FOAF allows groups of people to build up social networks without the need for a centralized database. Related schemas are **DOAC** (Description of a Career) for including information about education, working experience, publications, spoken languages, etc., **DOAP** (Description of a Project) for describing open-source software projects, and **SIOC** (Semantically-Interlinked Online Communities) for interconnecting online discussions through different channels like blogs, forums, and mailing lists. The SIOC project has developed linked data wrappers for several popular blogging engines, content management systems and discussion forums such as WordPress, Drupal, and phpBB Endnote. Based on such published information, one can, for instance, define metrics to determine social neighborhood and social reputation, which might be used in eScience [26].

Reasoner Benchmarking Ontologies

For the benchmarking of reasoner systems, specific ontologies are frequently used, while some have even specifically been constructed for this particular purpose. The **Wine ontology** (http://www.w3.org/TR/owl-guide/wine.rdf) is a rather expressive OWL ontology that makes use of sophisticated description logic constructs and is thus demanding for reasoners. The **Lehigh University Benchmark** (**LUBM**, http://swat.cse.lehigh.edu/projects/lubm/) [27] and the **University Ontology Benchmark** (**UOBM**) [28] define specific test ontologies in the university domain that are widely used. Also some of the ontologies mentioned above are often used for testing reasoner performance due to their size or complexity, such as GALEN or SNOMED.

13.2 Scientific and Technical Overview: Engineering and Methodological Aspects

Ontologies used in various applications differ, for instance, in terms of scope, size, or expressivity, that is, their degree of axiomatization. While it is often possible to reuse existing ontologies that fulfill all the requirements of a certain application, many practical scenarios demand for the acquisition of new ontological knowledge or the adaptation of previously given axiomatizations. Additional changes to the ontology might become necessary at runtime as the domain knowledge or user requirements evolve.

The corresponding modeling and maintenance tasks make high demands on scarce expert resources and the capabilities of human ontology engineers. In order to overcome this knowledge acquisition bottleneck, the manual ontology construction process must be supported by efficient software tools, including *ontology editors* or ontology development environments [29]. Ideally, these tools should be complemented by appropriate *ontology engineering methodologies* [30] – guidelines and best practices in ontology design, developed from practical experiences as well as from theoretical considerations of formal ontology. Such methodologies are indispensable in order to prevent modeling errors that might hinder the applicability of the engineered ontologies. Furthermore, by structuring the otherwise undirected ontology engineering process and hence preventing unnecessary iterations or overly long meaning negotiations, they can significantly speed up ontology construction. Nevertheless, the construction and refinement of ontologies remains a tedious and time-consuming endeavor unless it is assisted by automatic or semiautomatic knowledge acquisition methods. So-called *ontology learning* techniques aim at acquiring ontological entities and axioms from various kinds of data, including natural language text [31], multimedia documents [32, 33], and folksonomies [34].

This section will give an overview of tools and methodologies for ontology engineering, thereby focusing on description logics and, in particular, the Web Ontology Language OWL.

13.2.1 Manual Ontology Construction

The manual construction of ontologies is a challenging and time-consuming endeavor. According to [35], ontology engineering is "*the set of activities that concern the ontology development process, the ontology life cycle, and the methodologies, tools and languages for building ontologies*." *Ontology editors* or *ontology development environments*, which typically offer a variety of additional features and pluggable components, can significantly speed up the ontology development process, since they relieve ontology developers from the need to care about the syntactic representation of ontologies. In this section, a brief overview will be given (1) on how ontologies are constructed in real-life application scenarios, and (2) the kind of tool support that is available to actually create all the entities and axioms an average OWL ontology consists of.

13.2.1.1 Ontology Editors

Assuming it is already known that one would like to build an OWL ontology and have a rough idea of what it should look like in terms of scope, size, and expressivity, what would be the ontology editor of one's choice? While it is difficult to make a general recommendation, there are certain aspects, which make one or the other tool preferable in a given application setting. Some of these aspects are discussed in the following.

Visualization and Editing Paradigm

The effectiveness of ontology editors or engineering environments largely depends on the usability of the user interface and the editing or visualization paradigm it adheres to. An inappropriate type of visualization, that is, a lack of expressivity at the interface level, hinders ontology construction and might even provoke certain modeling errors such as erroneous disjointness assumptions or misinterpretations of domain-range restrictions [36]. While the visualization paradigm is typically determined by the requirements of a family of ontology languages, there is also a variety of editors for specific domains such as bioinformatics [37] or particular types of ontologies, for example, in terms of logical complexity. While tree-based visualizations, for instance, as provided by the majority of today's ontology editors, are most suitable for editing taxonomies (see Sect. 13.1) or other lightweight types of ontologies, the specification of arbitrary axioms or rules often demands for more powerful editing functionalities. To this end, more expressive graphical notations have been proposed, such as UML diagrams [38] or text-based interfaces, for example, based on controlled natural language [39–42].

Extensibility

Many ontology development environments such as Protégé or the NeOn Toolkit (see further below), therefore, consist of a plug-in infrastructure with an underlying API that can be used for extension by custom or off-the-shelf components for various ontology engineering activities, including modularization, alignment, querying, and semantic annotation.

Reasoning Support

If an ontology is constructed for the purpose of logical inference, for example, in a data integration scenario, minor modeling errors can have a huge impact on the overall usefulness of the ontology as they might lead to wrong conclusions or even cause logical inconsistency. In order to detect and possibly remove such errors, an ontology engineer should be provided with automated support for reasoning, that is, querying and classification, as well as for inconsistency diagnosis. Furthermore, depending on the intended use of an ontology and the required level of expressivity, additional language features such as rule support can turn out to be helpful in building the ontology.

Collaborative Ontology Development

The complex and time-consuming task of constructing an ontology usually involves several people with varying roles, among them *ontology engineers* and *domain experts*. In a centralized setting, these people, affiliated to different organizations and working in different places, jointly develop one common ontology. If the ontology consists of multiple parts or modules stored on different servers, for example, in a grid, and connected by mapping axioms, this ontology is typically called a *distributed ontology*. *Collaborative ontology engineering* is supported, for example, by the DILIGENT methodology (cf. Sect. 13.2.2), semantic Wikis [43], and plug-ins for various ontology editors such as Protégé [44].

Table 13.4 gives an overview of the most well-known ontology development environments for the Web Ontology Language OWL. All of these tools relieve ontology engineers from the burden of having to deal with a specific serialization format, helping them to avoid syntactic modeling flaws. However, they usually cannot prevent semantics-related errors caused, for example, by common misconceptions of the OWL semantics. In order to ensure a proper representation of the domain expert's conceptualization, guidelines and methodologies are required that assist ontology engineers in formalizing previously implicit knowledge in a meaningful way. The following Sect. 13.2.2 will therefore introduce some of the existing ontology engineering methodologies.

13.2.2 Methodologies for Ontology Engineering

An *ontology engineering methodology* is a set of procedures, guidelines, and best practices derived from real-world ontology development experiences (e.g., [45]) or theoretical

Table 13.4

Ontology development environments for OWL

Name	Organization	URL
Protégé	Stanford University	http://protege.stanford.edu
NeOn Toolkit	NeOn Foundation	http://www.neon-toolkit.org
Swoop	University of Maryland	http://code.google.com/p/swoop/
TopBraid Composer	TopQuadrant	http://www.topquadrant.com
SemanticWorks	Altova	http://www.altova.com

considerations of formal ontology [46]. It supports ontology engineers and domain experts, for example, in defining the structure of the ontology or in selecting formal and informal data sources to be reused. In recent years, the interest in such ontology engineering methodologies has grown significantly, not only thanks to the more and more widespread use of ontologies in practical applications, but most notably also because of the increasing size and complexity of those ontologies. Constructing large ontologies of sufficient quality with a minimum of human and financial resources poses one of the greatest challenges to the vision of the Semantic Web.

In the following, we will outline the basic structure of a methodological ontology construction process and describe some of the most well-known guidelines for proper ontology design.

13.2.2.1 Generic Methodology

The core of each ontology engineering methodology essentially comprises three steps [35, 47], which are explained in the following:

- *Requirements Analysis* – Usually, an ontology engineering process starts with a detailed analysis of the requirements that arise from the underlying application scenario. The ontology engineer or domain expert describes these requirements by means of an *ontology requirements specification* document, which serves as a basis for subsequent modeling activities and quality assurance, that is, later checks against the specification. For this purpose, the description of the requirements should contain information, for example, about the scope of the ontology (*competency questions*), its intended use, or the level of expressivity.
- *Conceptualization* – In the conceptualization phase, the ontology's content is represented in terms of semantic vocabulary and statements about the target domain of interest, which involves the choice of ontological entities and the formulation of axioms. Based on the aforementioned requirements specification, ontology engineers and domain experts try to achieve a common agreement upon the basic structure of the ontology, for example, by exchanging arguments to support their respective design decisions [48]. The result of this phase is an informal or semiformal specification of their shared conceptualization.
- *Implementation* – The explicit formalization of this specification in terms of a concrete ontology representation language is the final step of the core ontology engineering methodology. Choosing an appropriate ontology language most notably depends on the intended use of the ontology and the required level of expressivity. As will be seen in Sect. 13.2.3, the implementation phase can be supported by automatic approaches to ontology acquisition and reuse.

Furthermore, ontology development activities such as ontology *evaluation* [49], *versioning, documentation* or the reuse of existing, formal or informal resources (*knowledge acquisition*) might be required. In particular, it is often advisable to perform an alignment of the ontology with an existing upper-level ontology, whose axiomatization of the most fundamental ontological distinctions is often perceived as a valuable help in the overall quality assurance process.

Such an alignment can be achieved, for example, by applying *ontology design patterns* [50]. Ontology design patterns, similar to their counterparts in software engineering, incorporate best practices in ontology development by providing ontology engineers with templates for specific modeling tasks. *Figure 13.2* shows a simple content pattern [51] extracted from the DOLCE upper-level ontology [21], which can be used to represent the realization of an information object (see also http://ontologydesignpatterns.org). It could be specialized, for example, by a class *Novel* modeling an information object and a class *Book*, which represents its physical realization. Other types of ontology design patterns include, for example, logical patterns and correspondence patterns for ontology alignment [52].

13.2.2.2 Methodologies and Tool Support

The above-mentioned steps are part of more or less every ontology engineering methodology. Some of the most well-known methodologies facilitating the effective and efficient construction of ontologies are Methontology [53], OntoClean [46], and DILIGENT [54] (see *Table 13.5*). They all have been inspired by the first, general methodologies developed in the 1990s [61, 62]. Nowadays, the state-of-the-art comprises a variety of different methodologies for specific ontology development scenarios (e.g., distributed, collaborative ontology engineering [54]), and specific application domains, such as bioinformatics [63] or medicine [45].

Only a few of these methodologies have been integrated with major ontology development environments. However, the application of OntoClean, for example, is facilitated by several plug-ins, in particular for Protégé [64], WebODE [65], and OntoEdit [66]. These plug-ins facilitate the manual tagging of ontologies with OntoClean meta-properties and provide means for checking the consistency of an ontology according to the OntoClean constraints. An automatic approach to applying OntoClean has been proposed in [67].

Despite the existence of various ontology engineering methodologies, the construction of an ontology remains a labor-intensive, time-consuming, and error-prone endeavor if it

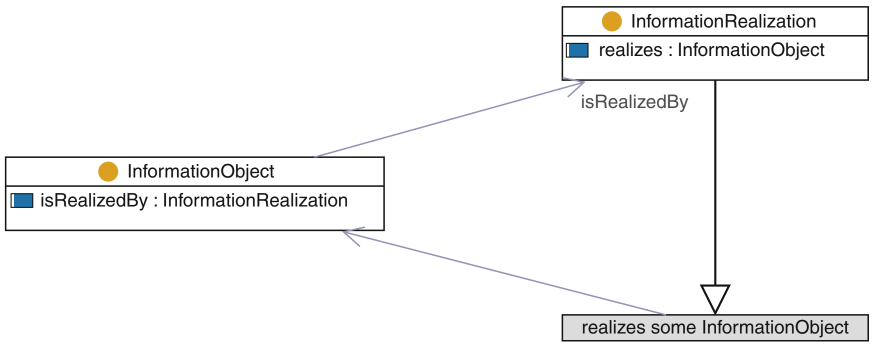

Fig. 13.2

Ontology design pattern: information realization, by Valentina Presutti and Aldo Gangemi (http://ontologydesignpatterns.org/)

Table 13.5

Ontology engineering methodologies

Name	Institution	Reference
Methontology	Universidad Politécnica de Madrid	Fernández-López et al. [53]
Ontology Development 101	KSL, Stanford University	Noy and McGuinness [55]
On-To-Knowledge	AIFB, Karlsruhe Institute of Technology	Sure et al. [56]
OntoClean	CNR and IBM Watson	Guarino and Welty [46]
UPON	CNR and La Sapienza, Rome	De Nicola et al. [57]
DILIGENT	AIFB, Karlsruhe Institute of Technology	Tempich et al. [54]
HCOME	University of the Aegean, Samos	Kotis and Vouros [58]
NeOn Methodology	Universidad Politécnica de Madrid	NeOn Deliverable 5.4.1 [59]
DOGMA	STARLab, Vrije Universiteit Brussel	Jarrar and Meersman [60]

is carried out entirely manually. The next section will therefore take a look at some automatic methods for knowledge acquisition and ontology generation that support this manual process, which are commonly known as *ontology learning* techniques.

13.2.3 Ontology Learning

The manual construction of ontologies is a time-consuming endeavor, which makes high demands on the skills of an ontology engineer. Although various editors relieve one from the burden of dealing with the OWL syntax by providing efficient support for creating, querying, and merging ontologies, and despite all the methodological guidelines available, the *knowledge acquisition bottleneck* still hinders the widespread use of semantic technologies. A possible solution to this problem is the use of *ontology learning* methods [68].

The term "ontology learning," coined in [68], refers to the automatic or semiautomatic generation of ontologies from various kinds of data sources including folksonomies [34], FOAF profiles [69], and multimedia documents [32, 33]. In its most commonly used sense, however, it is used to denote the acquisition of ontological knowledge from unstructured natural language text. This type of ontology learning, that could be described as "lexical" or linguistically motivated ontology learning, differs from logical approaches such as *concept learning* (ILP) or *relational exploration* (FCA) in terms of input data. While logical approaches typically derive new axioms from existing, formal, and explicit representations, lexical ontology learning has to face the challenge of dealing with large amounts of informal and often unreliable data. The following will mainly focus on the latter type of ontology learning – thereby omitting the distinction between ontology learning and *ontology population*, which is sometimes made in order to tell apart the acquisition of schema-level knowledge from the generation of instantiations and assertions (see Sect. 13.1).

13.2.3.1 Methods for Ontology Learning from Text

The target for ontology learning can vary among certain ontological entities or types of axioms to be acquired.

Concepts

Terms, that is, nominal or verbal phrases referring to linguistic concepts, are widely accepted as a means of labeling classes, individuals, and properties. One of the most fundamental tasks in lexical ontology learning, therefore, aims to identify terms or phrases that are relevant in a particular domain of interest (*term extraction*). Often the significance of term occurrences is determined by comparing their frequencies with statistics obtained for a reference corpus [70–72], or by considering structural information that is contained in HTML or XML documents [73]. For a good overview of term extraction methods, see [74] and [75]. In a second step, based on Harris' distributional hypothesis [76], these terms can then be grouped into clusters of synonymous or otherwise related terms describing a single concept [77].

Subsumption

The backbone of any ontology is constituted by a set of taxonomic relationships among concepts (or concept descriptions). Each of the classes can be defined intentionally, for example, by a descriptive label or its relationships to other classes, as well as extensionally by specifying a set of instances belonging to this concept. Since the core taxonomy of an ontology, independently of the underlying ontology representation language, is of crucial importance for the use of ontologies as a means of abstraction, most ontology learning approaches so far have focused on the formation of concepts and concept hierarchies. Accordingly, different approaches to learning subsumption have been developed.

The vast majority of these approaches rely on lexico-syntactic patterns, invented by Marti Hearst and commonly known as "Hearst patterns" [78]. These patterns, essentially linguistic expressions encoding both lexical and syntactic constraints on the textual context of a concept expression, have shown to be a helpful means to detect *hyponymy* relationships or categories of named entities. The following pattern, for instance, would match a noun phrase (or enumeration of noun phrases) followed by "*and other*" and finally, a second noun phrase denoting the super-concept of the first concept expression.

"*cars, ships and other vehicles*"
NP {, NP}* {,} {*and* | *or* } *other* NP

Additional patterns have been proposed by Ogata and Collier [79], Cimiano [80], and others. For an extensive comparative evaluation of various hyponymy patterns in a specific domain, see [81]. Common to all pattern-based approaches to hyponym extraction is the problem of data sparseness. Since occurrences of lexico-syntactic patterns are comparatively rare in natural language texts, most research nowadays concentrates on the Web as a corpus [82–85] – even though the enormous syntactic and semantic heterogeneity of Web documents poses new sorts of challenges. At the same time, approaches to the automatic generation of patterns [86–89] aim to increase the flexibility and effectiveness of ontology learning or information extraction systems. A different approach, specifically designed to support the acquisition of subsumption relationships, has been proposed by Sharon Caraballo and others [90–92]. It relies on hierarchical clustering techniques in order to group terms with similar linguistic behavior into hierarchically arranged clusters.

Instantiation

The tasks of learning instantiations and assertions, often jointly referred to as *ontology population*, are targeted at the acquisition of facts, that is, instance-level information. State-of-the-art approaches to determine, for example, class membership of individuals usually build upon computational methods for named entity recognition and classification. While some of them use lexico-syntactic patterns [80] similar to the aforementioned Hearst patterns, others exploit, for example, the distributional similarity of named entities and their semantic classes [93]. A method for extracting instantiations as well as assertions from natural language text has been proposed in [94] and relies on FrameNet as a specific kind of structured background knowledge. More details on ontology population can be found in Semantic Annotations and Retrieval: Manual, Semiautomatic, and Automatic Generation.

Assertion

Identifying relationships between named entities or individuals in a given ontology is usually considered a subtask of information extraction. A possible distinction can be made between *open* and *closed information extraction* approaches, differing in the degree to which they rely upon supervised learning techniques. While closed information extraction (e.g., [82, 89, 95]) typically presumes an explicit or implicit specification of the target relationship, hence requiring some amount of training data or handcrafted extraction rules, open information extraction [96] relies on merely unsupervised or semi-supervised approaches.

Domain and Range

Most approaches to the acquisition of non-taxonomic relationships (or object properties) are based on the analysis of verbs and their arguments as defined lexically by subcategorization frames [97]. A major challenge posed by this kind of approach to relation extraction is the identification of the right level of abstraction when it comes to determining the most specific restrictions holding for domain and range [98]. This also holds for more statistical techniques based on collocations [99] or association rules [100, 101] – even though the latter anyway demand for a certain degree of user interaction as the generated relationships most often lack meaningful labels (see Kavalek and Svátek [102] for some relation labeling experiments). A semiautomatic approach to refining logically complex domain-range restrictions of ontological properties has been developed by Rudolph and Völker [103, 104].

Disjointness

Disjointness axioms have attracted only little attention in the ontology learning community so far. On the one hand, the majority of approaches to ontology learning from text still concentrate on the generation of rather lightweight ontologies. On the other hand, it is very difficult to derive negative information from purely textual data. A classification-based approach presented in [105] therefore relies on heterogeneous types of evidence, in order to determine the disjointness of any two classes. Besides, logical methods based on inductive logic programming (e.g., [106, 107]) or formal concept analysis (e.g., [108, 109]) potentially generate disjointness axioms as a by-product of a more general learning algorithm.

Other Axioms

Even more difficult appears to be the automatic acquisition of arbitrary axioms. While initial blueprints already exist [110, 111], the quality of the learned axioms is still insufficient for most practical applications. Purely logical approaches could yield better results in terms of precision, but the sheer complexity of this task overtaxes the capabilities of many state-of-the-art concept learning approaches. Therefore, recent research and development efforts are aiming to facilitate the import of relevant axioms and modules from existing ontologies [112].

13.2.3.2 Tools and Implementations

In order to facilitate the integration of automatic or semiautomatic approaches into the overall ontology construction process, several *ontology learning tools* and *frameworks* have been developed in recent years.

An *ontology learning framework* is a platform that (1) provides multiple ontology learning methods, each of which completes one or more ontology learning tasks; (2) guides the user in selecting, configuring, and applying these methods; (3) integrates the results of each method into a common knowledge model; and (4) supports the user in reviewing and exporting this knowledge model. These requirements are to a certain extent fulfilled by the vast majority of today's ontology learning frameworks (see *Table 13.6*). *Figure 13.3* shows a screen-shot of [117], a framework for ontology learning from text that is available as a plug-in for the NeOn Toolkit (see Sect. 13.2.1). It is composed of different views for the configuration of the ontology learning process (i.e., corpus selection and workflow composition) and the tabular presentation of the results, which can be exported into the Toolkit's editor perspective as an OWL ontology.

In addition to the above-mentioned frameworks, most of them focusing on ontology learning from text, several tools have been developed in order to support the acquisition or refinement of more complex axiomatizations, for example, by means of formal concept

Table 13.6

Ontology learning frameworks

Name	Institution	Reference
ASIUM	INRIA, Jouy-en-Josas	Faure and Nédellec [92]
WEB → KB	Carnegie Mellon University	Craven et al. [113]
TextToOnto	AIFB, Karlsruhe Institute of Technology	Mädche and Volz [114]
Hasti	Amir Kabir University, Teheran	Shamsfard and Barforoush [110]
OntoLT	DFKI, Saarbrücken	Buitelaar et al. [115]
DOODLE	Shizuoka University	Morita et al. [116]
Text2Onto	AIFB, Karlsruhe Institute of Technology	Cimiano and Völker [117]
OntoLearn	University of Rome	Velardi et al. [118]
OLE	Brno University of Technology	Novacek and Srmz [119]
OntoGen	Institute Jozef Stefan, Ljubljana	Fortuna et al. [120]
GALeOn	Technical University of Madrid	Manzano-Macho et al. [121]
DINO	DERI, Galway	Novacek et al. [122]
OntoLancs	Lancester University	Gacitua et al. [123]

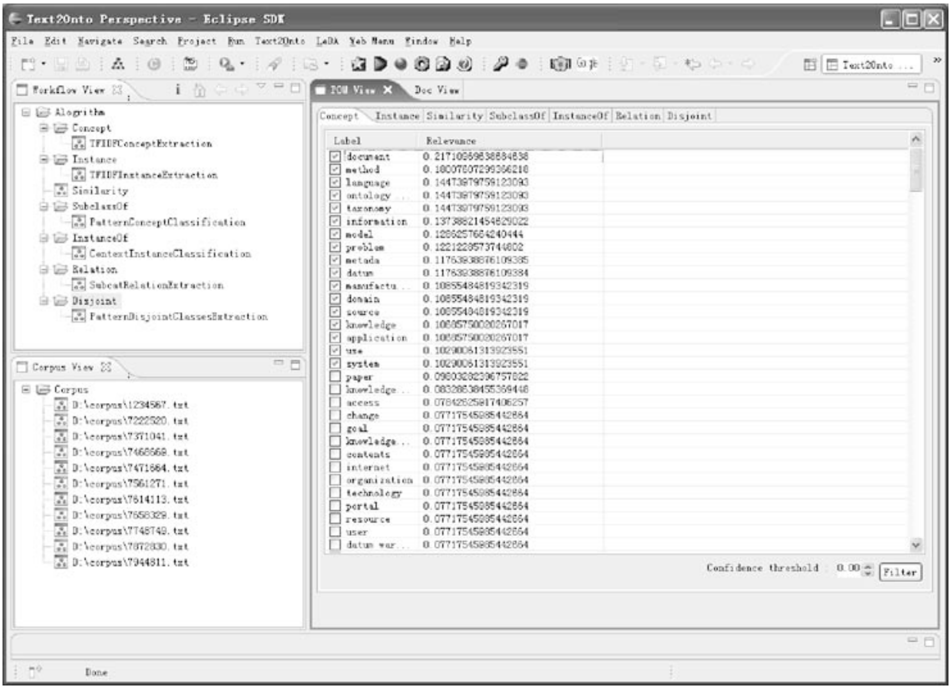

Fig. 13.3

Text2Onto plug-in for the NeOn Toolkit

analysis (FCA) or inductive logic programming (ILP). Some of the most well-known implementations are listed in Table 13.7.

Despite the progress made in recent years especially in terms of logical or lexico-logical ontology learning, the formal quality of ontologies that can be acquired by state-of-the-art ontology learning tools or frameworks is still insufficient for many practical applications. A tighter integration with ontology engineering methodologies and automatic means for ontology evaluation could help to avoid automatically introduced modeling errors and to reduce the required amount of post-processing for ontology learning. Sect. 13.2.4 takes a closer look at several past and ongoing efforts aiming to introduce best practices and theoretically well-founded quality criteria into automatic ontology acquisition.

13.2.4 Methodological Aspects of Ontology Learning

In recent years, a considerable amount of research has focused on the integration of methods for automatic ontology construction into general ontology engineering methodologies [125] as well as into specific methodologies for collaborative [126] or pattern-based [127] ontology design. At the same time, the ontology learning community has acknowledged the need for incorporating methodological guidelines and best practices into ontology acquisition approaches, for example, by applying ontology design patterns to learned ontologies [128]. Both directions of research, which could be referred to as *semiautomatic ontology engineering*, are among the most important topics for future studies on the Semantic Web. Only if one is able to combine the efficiency of automatic knowledge acquisition approaches with the skill and experience of human ontology engineers in a methodologically sound and effective way, one will eventually succeed in generating ontological resources on a large scale.

It is, however, a long way to go and the ontology learning community will have to show that it is heading in the right direction. Appropriate methodologies and benchmarks, for example, in the form of corpora or gold standard ontologies [129], for evaluating the effectiveness of ontology learning methods. A large fraction of these benchmarks should be tailored to particular applications or domains, thereby fostering the development of highly optimized ontology acquisition methods. Even though most of today's ontology

Table 13.7

Tools for logics-based ontology learning

Name	Institution	Reference
OntoComP	University of Dresden	Sertkaya et al. [124]
RELExO	AIFB, Karlsruhe Institute of Technology	Völker and Rudolph [104, 109]
DL Learner	University of Leipzig	Lehmann et al. [107]
YINYANG	University of Bari	http://www.di.uniba.it/~iannone/yinyang/

learning research still concentrates on general-purpose approaches, a constantly growing interest in the automatic acquisition of biomedical ontologies [122, 130], for example, motivates a diversification of the field. The development of specialized ontology learning and evaluation methods will also be driven by further application scenarios and opportunities, for example, in the field of product or service configuration, arising from an increasing expressiveness of learned ontologies.

The automatic or even semiautomatic generation of ontologies remains a challenging endeavor. However, there is hope that some of the guidelines and best practices derived from previous experiences in manual ontology design can find their way into automatic approaches to ontology learning and evaluation.

13.3 Example Applications

In the context of the Semantic Web, ontologies play a key role in that they provide the semantic vocabulary that semantic applications build on for the exchange and interpretation of Web data and information. Depending on the use-case scenario at hand, different features of ontologies are used in different ways within different types of applications for aiming at different goals, be it search, integration, or organization of knowledge.

This section will identify various kinds of usage of ontologies in Semantic Web information systems and relate them to typical application areas, before concrete usage examples with varying application domains will be presented.

13.3.1 Generic Functionalities of Ontologies in the Semantic Web

Regarding the wording used throughout this section, please note:

1. The term *Semantic Technologies* or *Semantic Web Technologies* shall denote the whole range of Computer Science and Artificial Intelligence methods and tools typically used and typically playing together in applications that rely on a formal ontology (or several ontologies, respectively) and on explicit, ontology-based metadata for information items or information systems – in order to enable better information search, integration, processing, or management, especially in distributed and open scenarios. Such Semantic Technologies comprise core aspects like ontology engineering and management, as well as metadata creation and management, but also contributing and underlying base technologies like natural language processing or automated reasoning.
2. Furthermore, terms like *Semantic Web applications*, *ontology-based applications*, *Semantic Technology applications*, *semantic applications*, etc., are used synonymously; that is, *Semantic Web–based* and *ontology-based* is meant interchangeably in this section, although, outside the Semantic Web scenarios, there are also other usages of formal ontologies, namely, in Multi-Agent Systems, or in Expert Systems.

3. Lastly, for the purpose of this application survey, the ideas of *Semantic Web* and *Corporate Semantic Web* are considered to be very similar. Of course, behind the firewall of a company, an Intranet application may be much easier to realize than a similar Internet application, from technical and from nontechnical points of view (trust, standards compliance, incentive systems, etc.); but, nevertheless, many company-internal information landscapes provide challenging-enough problems; some corporate Intranets today are bigger than the Internet was 10 years ago; companies must extend the scope of their electronic communication and collaboration sphere toward customers, suppliers, whole value creation networks; and the dynamics that many businesses have to cope with is really impressing; so altogether, Corporate Semantic Web is a challenging area, which may provide some clearer business cases than the whole Web can offer.

Before presenting *concrete* applications of ontologies in the Semantic Web, some *generic functionalities* of ontologies will be derived from definitional elements of ontologies as introduced above:

- *Formality* – This is the basis for a high achievable degree of automation when processing ontological knowledge or ontology-based metadata (and, as an effect, a potentially high level of automated "intelligence"). Obviously, the more heavyweight an ontology is (higher degree of formality), the more and more powerful automated and intelligent processing services are possible. Of course, a high degree of formality with its potential benefits must be traded in many scenarios against design and maintenance costs, and may be unnecessary or even inappropriate in others (e.g., in rather informally structured knowledge-organization tasks where the final knowledge consumer is a human, navigating user).
- *Explicitness* – Making modeling decisions explicit to a large extent, formally modeling them, or even modeling *redundant* aspects in an expressive language can be the basis for better human understanding where and how to use (or reuse) an ontology or ontology-based metadata, but it may also be the basis for the (semi)automatic mapping of different ontologies or for more automated inferences.
- *Consensus* – Consensus on modeling decisions is the most effective way to achieve data and system interoperability. So, harmonizing heterogeneous structures and viewpoints can be facilitated by a broad sharing of agreed-upon knowledge – in the extreme case, this leads to standardization. Of course, standardization is not always possible or (from a political, economic, or psychological point of view) achievable/desirable. But even where no consensus about specific domain- or task-related ontological knowledge can be achieved, the consensus on at least the basic knowledge representation formalisms (Semantic Web standards) makes interoperability already easier. Furthermore, formality and explicitness support the process of finding consensus as they allow one to communicate clearly what different parties are talking about. Lastly, an increasing number of publicly available ontologies will increase the degree of model reuse, which *implicitly* fosters consensus through de facto standardization.

- *Conceptuality* – This way of declarative modeling a domain's knowledge structures is the basis for all search-support functionalities offered to humans, for example, for navigation or for query disambiguation or reformulation (see below).

These characteristics contribute, to a different extent, to a number of potential basic benefits, which ontologies can offer to applications, namely (see *Fig. 13.4*): (1) *standardization of representation languages* (like RDF, RDFS, OWL); (2) *standardization of metadata schema* (like WSMO for Web Services); (3) *provision and/or standardization of domain-knowledge* structures; (4) *rich declarative modeling* languages; (5) *machine-processable semantics*, including automated reasoning facilities.

In addition to these basic benefits, which can be gained from using ontologies in general, for the specific case of *Semantic Web* ontologies, two further advantages can be mentioned, namely (6) that Semantic Web ontologies aim at being *incrementally extensible*, and (7) that Semantic Web ontologies and knowledge bases are designed for being built, extended, and maintained in a *distributed* manner.

The above listed basic benefits lead to a number of *generic functionalities* realizable through ontologies in (distributed, Web-based) Information Systems:

- *Knowledge and Information Organization* – Many knowledge and information management systems rely on informal, only partially structured, and hardly machine-processable representations of knowledge, such as books, personal notes, drawings, presentations, e-mails, or any kind of multimedia documents. These can be stored in digital libraries, document management systems, personal file systems, etc. Often, such storage systems are organized with the help of metadata that describe the stored

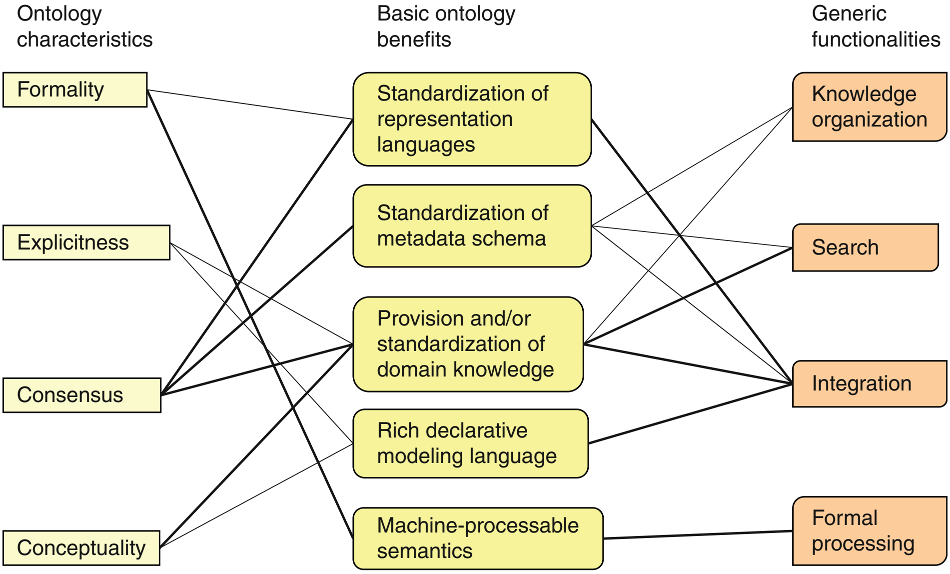

Fig. 13.4

Ontology characteristics lead to basic benefits and generic functionalities

knowledge items. To express such metadata, some systems employ a metadata schema in the form of an *information ontology*; moreover, content descriptions of knowledge items in metadata can refer to *domain ontologies*.

- Search – Almost all Semantic Web use cases deal, in one way or the other, with searching for Web-accessible resources. Ontologies can support and improve practically all aspects of information search (see Sect. 13.3.2).
- *Integration* – If data and information from more than one source shall be processed together, different data schemas must be linked together or (partially) be mapped onto each other. At least three cases can be considered: (a) If the data schemas to be mapped are expressed as ontologies, (*semi*)*automatic* mapping is facilitated through ontologies because rich and explicit, possibly redundant, models in a language with an unequivocal formal semantics provide a good starting point for that. (b) Even in the case of fully *manual* schema mapping: If schemas are expressed as ontologies (which typically come together with rule languages like SWRL or F-Logic), the rule-based formulation of schema mappings is an easy-to-use, yet powerful mechanism. (c) Independent from the issue of schema mapping *at query time*: There are also scenarios where it is wished to link data from different sources already at the information-provider side. This is facilitated much by Semantic Web standards, as they are designed for that purpose.
- *Formal Knowledge Processing* – Automated reasoning over ontology-based metadata and/ or ontological background knowledge is a functionality that may be used in either of the three other functionalities above, be it to achieve a higher degree of automation, to provide new functionalities enabled through executable knowledge representations, to deduce implicit knowledge, or to validate and verify knowledge formulated by the user.

These generic functionalities are jointly employed and instantiated in manifold ways in the more specific functionalities and classes of semantic applications collected in *Fig. 13.5*, which are further discussed below. Concrete application examples often combine aspects of more than one of these categories.

13.3.2 Applications of Ontologies in the Semantic Web

13.3.2.1 Semantic Search

Semantic search in Intranets and in the Internet is a very widespread application of ontologies which can be characterized by the following working definition: *Semantic search employs semantic technologies to support human or automated agents in the process of finding – in a given search context – from one or more (Web-based) information sources those information item(s) most appropriate for satisfying a given information need. Major aspects of this task are often (1) that the search system obtains a detailed and unequivocal interpretation of the semantic intent of the information need at hand; and (2) that the search system employs semantic metadata and/or domain-specific background knowledge about available information sources and information items for precisely and comprehensively answering this information*

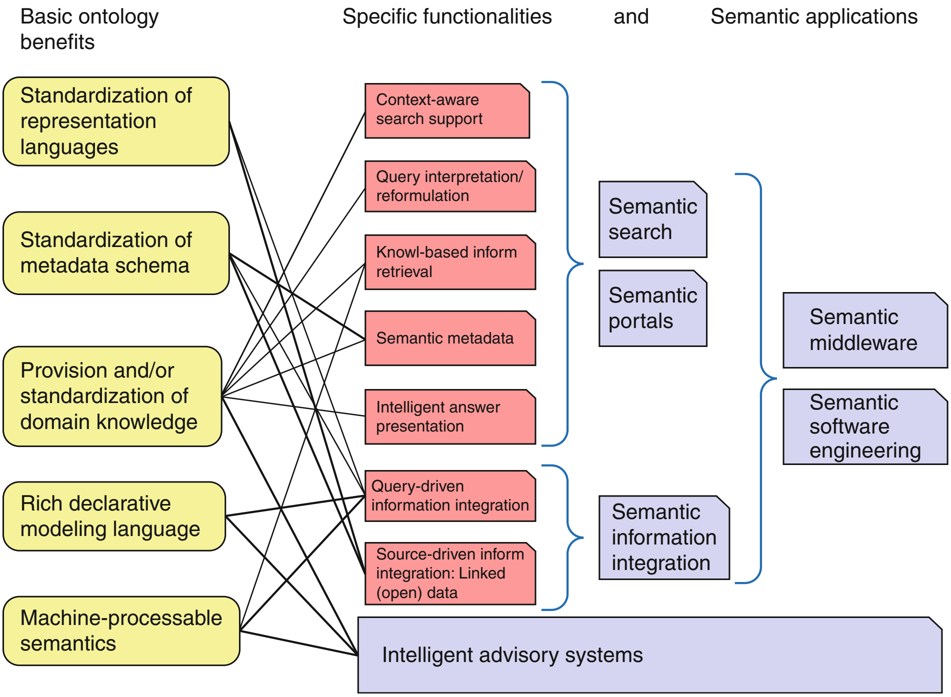

Fig. 13.5

Ontology benefits support specific functionalities and semantic applications

need. Often, some connected activities are also included, especially for helping the searching agent with making the best use of the found information, for example, by specific answer-set post-processing or result presentation.

A wide variety of approaches is subsumed here. The schematic illustration of an information-search process in *Fig. 13.6* shows points of application for ontology-based support:

1. *Context-aware search support*: The process starts with an agent being aware of a context-dependent information need; the agent articulates this information need in a form he or she is able to use. This form may be a list of search terms, a question in natural language, a document, which might be similar to the expected answer document, a SPARQL query, etc. Especially human agents who are not always aware that they have an actual information need can be supported by context-aware information systems: They detect the information need and can notify the user or even pose the query automatically to the search system – thus proactively delivering potentially relevant information [131, 132]. In all cases, *ontology-based personalization* may support information-need articulation with knowledge about the user's prior knowledge, major interests, specific wording, specific search constraints, etc. *Summary: Here, ontologies contain background knowledge and offer KR formalisms for user modeling, context modeling, activity rules, query-adaptation rules, etc.*

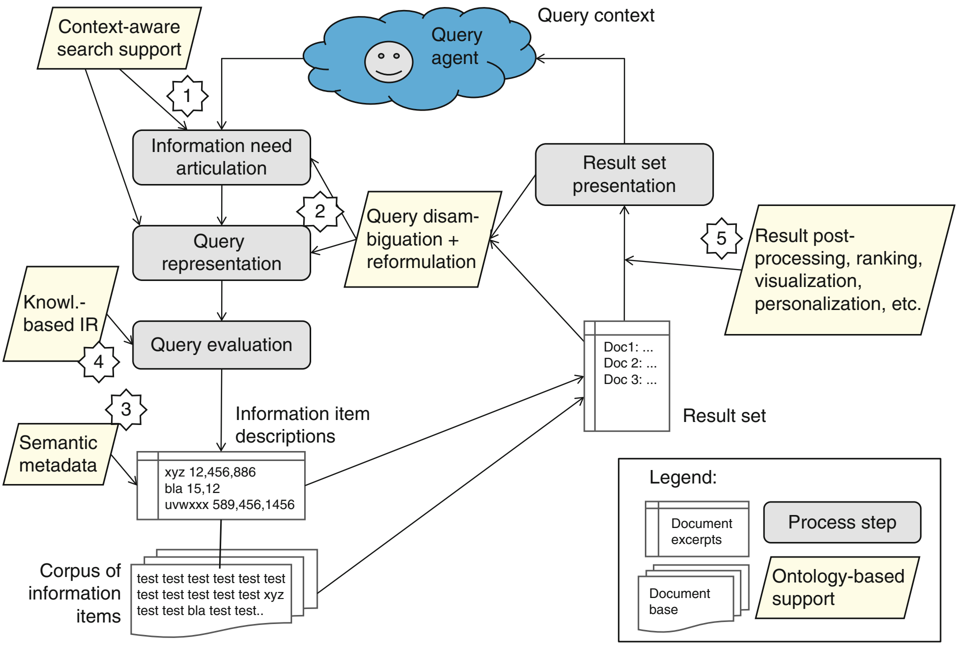

Fig. 13.6

Semantic search process

2. *Query disambiguation and reformulation*: If the articulated information need (for instance, a natural language question) and the query representation for the query evaluation back-end, for example, a Boolean query over document metadata) differ in their syntax, a translation process must take place. Even in the case of the same syntax for human query formulation and query evaluation, it may be derived from the ontological structures, from context knowledge or from analyzing the available information sources, that the formulated information need does not match the conceptual structures of information-item descriptions, or that it is ambiguous, too vague, too specific, or even inconsistent; then it must be refined or reformulated (see also [133]). In this disambiguation and semantic interpretation process [134], ontologies can (A) help to overcome natural language ambiguities through (A1) encoded knowledge about the relationship between natural language expressions and formal concepts and (A2) encoded content-specific background knowledge; they can (B) provide a fine-grained vocabulary for the formulation of very precise and unambiguous as well as very complex questions (e.g., using attributes and relations). In Fig. 13.6, the arrows from phase (5) back to the disambiguation and reformulation step indicate that also too many, too few, or bad answers can give reason to change the query posed to the system. This may happen interactively with the user, or (heuristically) automated with query-reformulation rules. In Information Retrieval (IR), such approaches are well-known as *query expansion*, whereas here, the term *query reformulation* is preferred because the query may be extended, refined, or completely

changed. If the knowledge space exploited for query formulation is given through a domain ontology, one would talk about *concept-based query reformulation* [135], in contrast to a *lexicon-based* query expansion, which only considers *wording aspects*. *Summary*: *Altogether, the domain ontology provides the knowledge space within which the query can be varied.*

3. *Knowledge-Based Information Retrieval*: At the heart of each search solution stands the query evaluation subsystem which matches the formal representation of an information need against the formal representations of information-item content in order to identify those information items (text or multimedia documents, database records, etc.) with the highest probability of satisfying the information need at hand. For doing that, several IR paradigms have been devised, the *Vector Space model* [136] working on textual keywords being the most prominent one. When information items are represented by declarative metadata records, the keyword-based paradigm can be replaced or extended in the most simple case by the *Boolean retrieval model* [136]. If semantic metadata are fully expressed in or do at least refer to logics-based representations in a Semantic Web language, one can use a *logic-based IR approach* [137]: Here, the relevance of an information item for a given query is logically inferred on the basis of given metadata facts and ontological background knowledge (by deductive or abductive inference; often also taking into account the uncertainty and vagueness of IR probabilistic, possibilistic, or fuzzy logic; for systems based on description logics, there are also approaches implementing retrieval as *classification* inference [138]). Concrete semantic search implementations often choose a hybrid approach combining text-based and metadata-based retrieval. Another form of knowledge-based IR, which is not logics-based, but can exploit ontological background knowledge, is *similarity-based retrieval* coming from Case-Based Reasoning [139]: Here, both the information items and the query are represented as sets of attribute-value pairs; the expected utility of an information item for a given query is assessed through a weighted sum of local value-similarities per attribute, compared between the stored item and the query. Ontologies can here define the ranges for individual attribute values and then provide the basis for local similarity assessment. *Summary*: *Here, ontologies provide the knowledge base for assessing relevance of information items, their formal semantics is the starting point for declarative retrieval models, and they can provide background knowledge for nontrivial inferences bridging larger conceptual distances between query concepts and concepts in the descriptions of information items.*
4. *Semantic metadata*: Describing information items or underlying information sources through semantic metadata means to make sets of statements that instantiate an information ontology (aka metadata schema); these metadata statements may refer to one or more domain ontologies for describing the content topics an information item is talking about. Mechanisms for including links to domain ontologies (or other rich knowledge models) for sophisticated content description are already foreseen in many metadata standards like the Dublin Core for general electronic documents, or the IEEE Learning Object Metadata standard (LOM) for eLearning resources. Besides the possibility of giving expressive content descriptions, rich metadata can also bring

further benefits: (a) they allow one to describe real metadata, that is, aspects of an information item, which are not *contained* in this item, for example, the creation context of a lesson learned from a project, the expected prior knowledge for an eLearning lesson, etc.; (b) they allow one to refer to finer-grained pieces of knowledge than the whole information item; for example, they could represent individual chapters of a book; (c) vice versa, they can also represent more coarse-grained content objects, for example, a whole document collection or a set of complementary, interlinked eLearning resources (motivational examples, a mathematical theorem, its corollaries, example applications, and related exercises); (d) declarative metadata allow one to represent non-text information items, like multimedia content; (e) through standardized, homogeneous metadata, information items from different heterogeneous sources can be brought together, put into the same context, and interlinked. *Summary*: *Regarding semantic metadata, Semantic Web ontologies provide rich and machine-processable representation languages, they can standardize metadata schemas and provide standardized domain knowledge for content description.*
5. *Post-processing of search results*: If a number of potential answers have been retrieved, several kinds of post-processing may happen, which can be supported by ontologies: (a) results can be ordered and *ranked* according to declarative rules; (b) when retrieving *informal* documents (in contrast to *factual* knowledge), information extraction algorithms can be applied for gaining factual knowledge from the documents; this knowledge can be stored in ontological data structures and be *further processed* with knowledge-based methods; (c) especially in the case of large answer sets or complex information spaces, some aspects of the answer set are *visualized* for further human inspection and browsing; for instance, documents can be arranged according to their relevance to domain topics or visualized with their interrelationships as semantic networks; (d) also the answer post-processing and presentation can be subject to personalization and context-specific adaptation. *Summary*: *In result post-processing, ontologies provide the data structures for formal knowledge representation, the specific post-processing knowledge, or the domain-knowledge structures as the backbone for visualization, respectively.*

Please note two simplifications of the schematic semantic search process:

1. Only *one* information source is considered; in general, many sources may be available. Then, ontology-based metadata may be used to select the most appropriate source(s) for a given query. It can happen that the query evaluation subsystem must then dispatch several partial queries to the respective information sources, possibly in different query languages, and must reintegrate the results from different sources – which essentially amounts to the use case of query-based information integration (see Sect. 13.3.2).
2. Only *query-based* information access is discussed. But especially when human agents drive the scenario and when a relatively unknown information domain is to be explored, a *navigational* access may be more suitable. Combinations of both are also possible.

A general remark: In today's still standard case of full-text indexes for information-item representation (i.e., when neglecting all possibilities of ontology-based metadata), the query evaluation would typically rely on Vector-based IR methods. Also with such a fully conventional search engine in the back-end, the above explained approaches (1), (2), and (5) will perfectly work, exploiting the ontological knowledge to enhance the user interface, without a need of changing the legacy systems. Just the query evaluation (3) itself can only be radically improved if the information-item descriptions are based on semantic metadata (4).

13.3.2.2 Semantic Portals

A Web Portal is a unique place in the Internet or a corporate Intranet to gather and present information from diverse sources in a unified manner; typically collecting and syndicating content (streams) about one specific topic, domain, region, or company, facilitating the work of one topic-oriented community (community portal), or the collaborative effort of a team with a dedicated task (project portal). Content types may include news and up-to-date information streams, e-mail, as well as (multimedia) documents. Web portals often provide a consistent look-and-feel for heterogeneous input, with access control and interfaces for multiple applications, for example, information push services (like RSS feeds) or comfortable information access with mobile devices. Based on [140], the following typical functionalities can be listed:

- *Information Supply* – Users have easy means to submit information and make contributions to their community(ies).
- *Information Management* – Portal administrators can easily integrate new (static or dynamic) information sources, keep the content consistent, change layouts, etc. This can include the (automatic) establishment of *links* between content items, which are not existing at the level of the individual content items.
- *Information Browsing* – Domain-knowledge structures are the basis for navigation menus, faceted browsing, information visualization, etc.
- *Information Search* – Unified search over heterogeneous content is provided.
- *Personalization* – Individual configurations for layout, information-delivery modalities, content selection, etc.

Compared to a conventional Web Portal, a *Semantic Portal* can be characterized by: (1) a domain ontology used as the central, harmonized topic structure for knowledge organization, navigation, and visualization; (2) semantic search mechanisms; and (3) Semantic Web languages for internal data management – which facilitates declarative approaches for further functionalities like personalization [141] or consistency checking of content. **OntoWeaver** [140], **Ontoviews** [142], or **SEAL** [143] are well-known Semantic Portal frameworks; see [144] for an overview.

Below, some examples for semantic search and semantic portal solutions are listed:

- References [145–147] analyzed and compared numerous academic and commercial semantic search tools with respect to some of the functionalities explained above (see also http://swuiwiki.webscience.org/index.php/Semantic_Search_Survey).

- There are many commercial tools for *semantic enterprise search* implementing some of the aspects discussed. For instance, ontoprise's Semantic Miner realizes many of the aspects under (2), (3), (4), and has a number of operational installations in large enterprises (see http://www.ontoprise.de/).
- Reference [148] presents a *corporate-search solution* for large, heterogeneous document collections dealt with by engineers in Rolls-Royce plc. Metadata can be manually edited, semiautomatically created, or extracted from legacy documents in a fully automatic manner. All document metadata are stored in OWL and RDF triples. The retrieval approach is hybrid and combines text-based with metadata search.
- Some examples for *similarity-based retrieval* can be found in the area of competence and skill management when human skill profiles are compared with, for example, job or actual problem descriptions, for instance, in Expert Finder systems [149].
- Many projects build portals for *cultural heritage* information (like descriptions of museum exhibits, content of national archives, or libraries [150, 151]). Besides issues like faceted search or metadata interoperability, these scenarios typically involve time and spatial aspects in metadata representation and querying – which is a special challenge for ontology-based approaches. References [152, 153] describe large-scale demonstrators for aspects such as result clustering and semantic recommendation (see also › Multimedia, Broadcasting, and eCulture).
- Numerous authors discuss semantic annotation of *biomedical* resources and subsequent search and question answering [154, 155]. **HealthFinland** is a semantic portal for health information in Finland where a Finnish ontology for laymen searching health information is constructed from several input ontology sources such as the Medical Subject Headings (MeSH) in SKOS format [156]; the content and metadata-creation workflow and tool support are presented as well; finally, the user services and interface design as well as their evaluation are discussed.
- In the *AgentDysl* project [157], retrieved eLearning resources for an electronic training system for dyslexic learners, are at real time arranged and adapted such that they best support the individual learning style and the current mood and psychological status of the learner, thus implementing a rule-based, context-specific result post-processing.

13.3.2.3 Semantic Information Integration

Data or information integration as the basis for data reuse, for query answering from multiple (heterogeneous) sources and for interoperable software systems in organization-internal (Enterprise Application Integration, EAI) or cross-organizational settings (eBusiness, Business-to-Business Communication, B2B) is a long-standing goal of Computer Science and a major motivation for the interpretation of the Semantic Web as a *Web of Data*. Jointly exploiting structured or semi-structured information from multiple sources has been researched in the Database community, under labels such as federated information systems, federated databases, etc. In principle, there are two major approaches: (a) the traditional way of looking at the topic, where data or systems, which were not designed for interoperability shall be queried in one interface and information pieces from different existing information sources must be found, combined, and aggregated – this is called *Query-driven Information Integration*; (b) the relatively novel approach of *Linked* (*Open*) Data where people intentionally use Semantic Web standards in order to foster the reuse and linkage of their data in manifold application contexts. This is a bottom-up approach, that is, *data-source driven*.

Query-Driven Information Integration

If data from different systems shall be integrated, a number of different kinds of heterogeneity can occur, which must be dealt with ([158] list some dimensions of heterogeneity). To this end, several integration approaches and architectures are possible; in the Semantic Web area, complying with Wiederhold's wrapper-mediator-architecture [159], most widespread is (according to the terminology in [160]) the *hybrid ontology approach*: A local ontology is built for each data source, representing its database schema in a standards-compliant, knowledge-rich manner. There is one global vocabulary or ontology, which is used at the query side where all local ontologies are mapped onto.

Figure 13.7 illustrates the general idea, as well as points of application for ontology support in data integration, based on [161]:

1. *Query-answering support*: A shared central vocabulary provides a *comprehensive and unifying conceptual view* on the application domain, independent from partial and

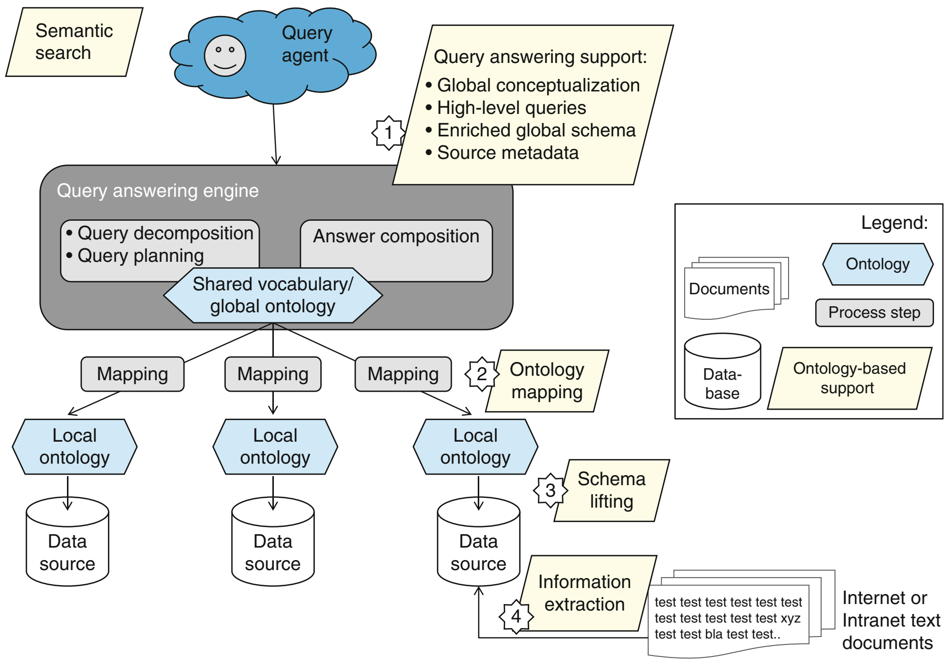

Fig. 13.7
Ontologies in query-driven information integration

heterogeneous, maybe implementation-driven, local schemas, or ontologies at the level of the individual information sources. This allows one to formulate *high-level queries* without knowledge about the different back-end data sources. Indeed, it is not even necessary to know which data sources exist, because the user can employ the global ontology as the only reference point for queries. It is also possible that the query answering engine uses metadata about data sources for *query decomposition*, *query planning*, etc. Such metadata may contain information about the content coverage of certain data sources, about data-source availability, about quality of content, or about costs associated with data access. Furthermore, *answer composition* from individual partial query results can be carried out declaratively, for example, in a rule-based manner. Lastly, at the level of the global ontology, *model enrichment* can happen, that is, that additional knowledge (in database terms, e.g., by a view mechanism) is inferred from data from different sources such that the global access layer may contain facts, concepts, and relationships, which do not exist in any *single* data source, but only virtually when looking simultaneously at the *whole* data landscape. A problem of answer composition from multiple sources is called *data matching* [162]: Identify that information from different sources refers to the same real-world object. Here, an ontology can contribute background knowledge and can be the hook for sophisticated reasoning about identity.

2. *Ontology mapping*: If both local data schemas and global schema are represented as ontologies, *declarative schema mediation* is possible: Mappings can be found easier because all schemas are represented in one standardized language; mappings can be represented in a rule-based manner – which is easier to formulate and maintain than procedural representations; and the manual formulation of mappings is often supported by visual editors. Using rich ontological knowledge, the *finding of mappings can be partially automated*. Creating ontology mappings is discussed in a large research community and has produced many valuable results [163, 164].
3. *Schema lifting*: As said above (under 2), the first step for ontology-based information integration is to express all local data schemas in the same ontology language. This step – in particular, lifting the data schema of a relational or an XML database – can also be (partially) automated by Semantic Web technologies (see, e.g., [161]).
4. *Information extraction*: Up to now, only discussed is the case that the data sources to be integrated are (semi-)structured databases like relational or XML databases. But, in general, all considerations still hold true if the back-end (partly) consists of less structured, informal information sources (like HTML pages with free text); these can be analyzed with NLP methods and – to some extent – be formalized, typically, by *information extraction* (IE) techniques, which identify specific kinds of information in texts and create structured representations for them. The result of such extraction tools can then either be stored persistently in a database or be created ad hoc at query time and queried like a structured data source (see also Semantic Annotations and Retrieval: Manual, Semiautomatic, and Automatic Generation). If such IE techniques are employed, the results may be directly stored as ontological assertions. An example is the MUSING system [165] which uses ontology-based IE to fill an integrated knowledge base from distributed, unstructured Web resources.

More information on the status of semantic information integration can be found here:

- Reference [166] is a comprehensive survey of semantic integration approaches, including the classification of many systems. Reference [162] gives a survey of semantic integration approaches seen from the database community, including a list of open problems.
- Reference [167] shows with many examples how to do rule-based information integration with F-Logic in the automotive area for *company-internal* purposes, whereas [168, 169] discuss *cross-company* information exchange in the automotive retail domain, based on their suggested **STAR** RDFS reference ontology for automotive retail.
- The **NeuroBase** project [170] is an example application for integrating heterogeneous information from different experimental sites, hospitals, and research centers in the area of cognitive neurosciences. It implements a standard wrapper-mediator architecture with a central unifying ontology.
- References [171, 172] present the **Crossvision Information Integrator**™product of Software AG, which almost completely supports the architecture in *Fig. 13.7*.

Source-Driven Integration: Linked (Open) Data

The Linked (Open) Data initiative (LOD) ([173], see also Semantic Annotation and Retrieval: Web of Data of this handbook) is a relatively new endeavor, which is based on a few best practices and technological principles for publishing and connecting structured data on the Web, extended by machine-readable metadata. The technical realization is based on the thorough use of URIs, the compliance with open Web standards like RDF and SPARQL, and the extensive use of links between data. Already in the recent few years, a global data space evolved containing 4.7 billion RDF triples interlinked by 142 million RDF links (figures from May 2009) – the "Web of Data" – connecting data from diverse domains such as people, companies, books, scientific publications, films, music, television and radio programs, genes, proteins, drugs and clinical trials, online communities, statistical and scientific data, and reviews (see *Fig. 13.8* from http://richard.cyganiak.de/ for an overview of data sets included until July 2009). Applications comprise, for instance, *Linked Data browsers*, which allow users to start browsing in one data source and then navigate along links into related data sources; or *Linked Data search engines* that crawl the Web of Data by following links between data sources and provide expressive query capabilities over aggregated data, similar to how a local database is queried today. Reference [173] enumerates a number of *Linked Data publishing* tools for both content in RDF stores and in non-RDF legacy databases. SparqPlug [174] is a service that enables the extraction of linked data from legacy HTML documents on the Web that do not contain RDF data. The service serializes the HTML DOM as RDF and allows users to define SPARQL queries that transform elements of this into an RDF graph.

A prominent example for an LOD application is DBpedia Mobile [175], a location-aware LOD browser running on a mobile device like the iPhone. DBpedia Mobile supports a tourist exploring a city. Based on the current GPS position of the mobile device, DBpedia

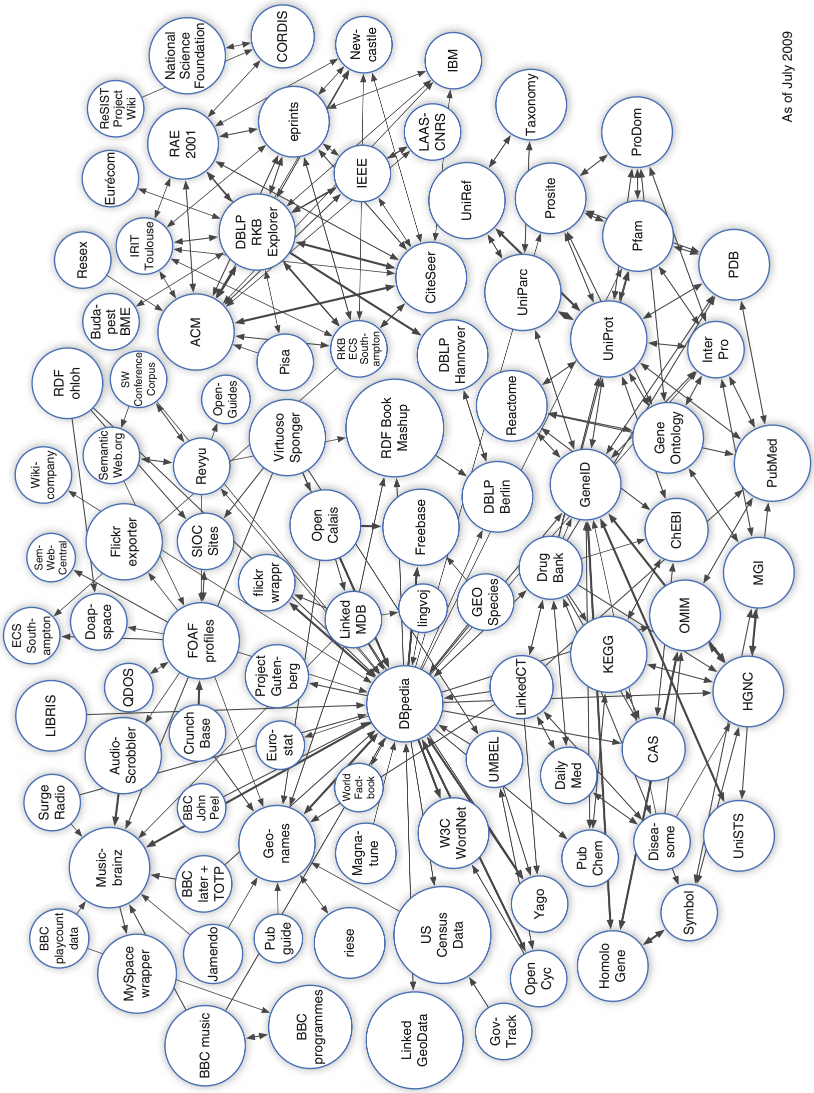

Fig. 13.8
Linking Open Data cloud, by Richard Cyganiak and Anja Jentzsch (http://richard.cyganiak.de/2007/10/lod/)

Mobile provides a location-centric mash-up of nearby locations from DBpedia, plus associated reviews from Revyu, and related photos accessed through a linked data wrapper around the Flickr photo-sharing API. DBpedia Mobile also enables users to publish their own current location, pictures, and reviews as Linked Web Data. Published content is not only described through geographic coordinates, but is also interlinked with a nearby DBpedia resource.

The LOD approach thus relies on a *pay-as-you-go data integration* approach [176] based on a mixture of using common vocabularies together with data source–specific terms that are connected by mappings as considered necessary.

Regarding the role of ontologies for LOD, simple information and domain ontologies like FOAF, SIOC, SKOS, DOAP, vCard, Dublin Core, OAI-ORE, or GoodRelations are used wherever possible. If new terminology is defined, it should be made self-describing by making the URIs that identify terms Web dereferencable. This allows clients to retrieve RDF Schema or OWL definitions of the terms as well as term mappings to other vocabularies. Linked data should be published alongside several types of metadata, in order to increase its utility for data consumers and quality, including provenance meta-information about data-creation time and procedures.

A remark on the practical relevance of the LOD initiative: One might wonder whether the idea of opening own databases for the public is naive or altruistic and what should be the economic incentives for doing so in business. There is a really huge noncommercial sector of information producers, like public authorities in eGovernment and eParticipation (this includes, e.g., the area of environmental information services or statistics offices), the whole scientific area, the area of NGOs (nongovernmental organizations) with altruistic motivations like charity that often have a strong interest in getting their information public, and finally the large area of information without active enforcement of copyrights, for example, historic documents. Moreover, even in the business sector, there are sometimes noncompetitive areas (e.g., research collaborations between companies, which search their Unique Selling Propositions in the later commercialization of research results).

Moreover, the LOD methods and principles can, of course, also be applied *within* an enterprise. For instance, the British Broadcasting Corporation (BBC) uses linked data internally as a lightweight data integration technology [177]. Formerly, BBC's numerous stations and channels used separate content management systems. Recently, BBC started to use linked data technologies together with DBpedia and MusicBrainz as controlled vocabularies to connect content about the same topic residing in different repositories. This content is augmented with additional data from the LOD cloud.

13.3.2.4 Intelligent Advisory Systems

An Expert System (XPS) [178], or more general, a Knowledge-Based System, is a computer program that simulates the judgment and behavior of one or more human experts who have expert knowledge and experience in a particular field. Typical problems addressed are diagnosis, configuration, planning, process monitoring, teaching, design, data analysis, or forecasting; application domains comprise medicine, mechanical, electrical, and civil engineering, manufacturing, chemistry, biology, geology, law, and many more. Typically, such a system contains a declarative knowledge base containing accumulated experience and a set of rules for applying the knowledge base. From a purely technological point of view, XPS have been mature since the late 1980s, to achieve a problem-solving performance comparable to a human's solution in quality, in many sophisticated cases. Nevertheless, XPS never had the big breakthrough expected at the time. The reasons for this include – besides exaggerated expectations and promises, and also technological problems, which can be considered solved in the meanwhile (performance, integration) – (a) that from an economic point of view, the creation and maintenance of XPS were far more expensive and complicated than acceptable in many operational settings, and (b) that it is always psychologically difficult to completely delegate complex decisions in difficult situations to a machine. So, modern applications of XPS technology often realize the idea of *intelligent advisory systems which do not fully automate a complex decision, but instead help a human user to find a decision, by, for example, (1) delivering important information (aka intelligent information retrieval), (2) helping to analyze complex or voluminous data and information streams, (3) checking complex constraints (aka automated critiquing component), or (4) suggesting partial problem solutions to the user for further human inspection and processing*.

Aspects (1) and (2) above refer mainly to semantic search and semantic information integration, whereas aspects (3) and (4) can be well implemented using "traditional" XPS techniques (which are nowadays increasingly amalgamated with Semantic Web languages and technologies). Settling upon state-of-the-art Semantic Web approaches facilitates systems interoperability and promises more cost-efficient systems engineering through the reuse of existing ontologies. Examples for the practical usage of intelligent advisory systems are listed below:

- The **Hospital Care Watch** [179] is a patient-management assistant prototype using an ontology about hospital care concepts (including hospital activities, procedures, and policies, and insurance policies, as well as medical knowledge per se) and a set of rules – for tracking the implications of medical decisions taken by physicians and other medical professionals within the context of guidelines and regulations, in order to avoid errors.
- References [180, 181] explore the combinations of ontologies and fuzzy inferences in medical application areas, such as respiratory waveform classification or diabetic food recommendation.
- The **IASO system** [182] is a mobile application which provides in a context-specific manner knowledge about a patient to a doctor outside the hospital. The system is based on three OWL ontologies formalizing, respectively, (1) patient histories in the hospital information system, (2) the description of the current patient situation, and (3) the context-specific relevance connecting elements/subsets of (1) and (2). The system is implemented in Java with the OWL API and employs the Pellet reasoner.

- Reference [183] describes an application of the ontology- and rule-based **Semantic Guide** product for supporting the customer service at a manufacturer of industrial robots. The implementation is based on ontoprise GmbH's **Ontobroker** F-Logic reasoning engine.

13.3.2.5 Semantic Middleware

In the recent decade, computer software and system architectures are increasingly moving away from monolithic, "one-program-on-one-computer" approaches toward distributed computing – putting together partial contributions from different, loosely coupled subsystems, often connected and communicating through standard Internet protocols. Manifold concrete realizations have been developed, from a virtualization of processing or storage system (as realized by *Grid or Cloud Computing*), to the composition of complex workflows from simpler, Web-accessible services (as realized by *Web Services and Service-Oriented Architectures*), or even to giving up the traditional notion of control in a program such that system behavior is emerging from the dynamic communication between partly autonomous units (as realized in *Multi-Agent or Peer-to-Peer Systems*). In all these paradigms, end-user interface and end-user application are decoupled to a different extent from concrete machines and operating systems, sometimes also from more abstract computing resources. So, all technical implementations of the aforementioned paradigms can be considered *middleware* [184, 185] components. Practically, all these middleware approaches face (to a differing extent) the following challenges at runtime:

1. *Find* the most appropriate available subsystem (Web service, peer, Grid resource, etc.)
2. Establish close *communication* between loosely connected elements (message exchange)
3. Realize an overall system *control* without a strong control paradigm

Obviously, challenge (1) might be improved by *semantic search*, (2) by *semantic integration* and mediation, and (3) by *reasoning* or at least some form of rule-based, declarative programming inside or outside the individual subsystem. Consequently, all paradigms have already been combined with semantic technologies:

- *Semantic Web Services* – Certainly the most active area to be discussed here [17, 186, 187]. A number of competing approaches for knowledge-rich Web service metadata have been suggested (see above: Service Description Ontologies) and been proposed for standardization. This falls under the above-introduced application class of *semantic metadata* in the area of semantic search. And, indeed, also manifold *knowledge-based IR techniques* for *service discovery* have been developed, comprising structural ontology matching [188], deductive retrieval using F-Logic inference for WSMO services [189], description logic inferences based on OWL-DL for SAWSDL services [190] or hybrid approaches (e.g., [191, 192] combine ontology-based type matching, logical constraint matching, and syntactic matching for WSMO and OWL-S service profiles, respectively, in a mixed similarity measure). The benefits from using ontologies lie in the formality and expressiveness of the representation language, and in the standardization of domain knowledge, for example, about products and services. For the task of *Web service composition* [193] to combine simple functionalities for achieving more complex ones, including the adaptation of interfaces between services, which might not perfectly fit together, many ontology-based solutions have been devised, too. These sometimes employ well-known *ontology mapping* techniques, and often are just declarative approaches, which rely on the fact that Semantic Web Services are described in a high-level, rich manner on the basis of a formal language. For instance, [194] describes an interactive service composition tool for WSMO service profiles, [195] uses linear-logic theorem proving for processing DAML-S service profiles, [196] applies abductive reasoning with constraint relaxation, based on former work in AI planning [197], and [198] combines description logic reasoning with situation calculus from AI planning. Please note: Techniques developed for Semantic Web Service Management can more or less directly be reused for the organization-internal management of workflows or other activity descriptions. For instance, [199] describes an interactive workflow-composition tool for configuring scientific workflows in seismic hazard analysis (earthquake research) based on similar principles as described above; [200] model biological processes with WSML and are then able to do sophisticated pathway analyses (explorative, verification) of biological process chains. See Semantic Web Services for more details on this area.

- *Semantic Grid* – Grid computing is a form of distributed computing based on a "virtual super computer," which is composed from a huge cluster of loosely coupled standard computers [201]. These cluster computers may be heterogeneous and may be geographically distributed. It has its major applications in science, especially in the typical application fields of high-performance computing, such as simulation in meteorology, in pharmacy, geology, etc., but also in some commercial fields, for example, in the automotive or in the finance area. The vision of the Semantic Grid [202] semantically represents Grid-resource metadata with RDF such that, in principle, pretty much the same benefits are possible as described above for Semantic Web Services.
- *Semantic Peer-to-Peer* – In a peer-to-peer (P2P) distributed network architecture [203], participating computers make a part of their resources (storage space, network band-width, processing power) *directly* (without intermediary hosts or servers) available to other peers. So, each peer is both a consumer and supplier of resources. Obviously, indexing and resource discovery are challenges in large P2P networks if one wants to find the most appropriate peers for a certain purpose and avoid huge communication traffic in the network. P2P systems often implement an application layer overlay network on top of the physical network topology, used for indexing and peer discovery. Reference [204] introduced the term *semantic overlay network* for a logical peer organization which is based on content-oriented classification hierarchies. Reference [205] presents a schema-based peer-to-peer overlay network that facilitates efficient lookup for RDF-based information in dynamic environments, which uses, among others, the semantic clustering of peers based on RDFS ontologies as one routing

optimization technique. The approach is applied in the scenario of building context-aware applications in smart spaces [206]. GridVine [207] may have been the first semantic overlay P2P system with an explicit focus on high scalability as well as semantic interoperability through schema reconciliation; the system is based on local RDFS schemas and uses OWL to encode schema mappings. Bibster [208] was an early application of semantic P2P overlay systems in the area of sharing bibliography entries; it uses lightweight RDFS versions of SWRC as an information/domain ontology and the ACM Topic Hierarchy as a domain ontology/vocabulary; Sesame is used as a triple store.

- *Semantic Cloud* – Cloud Computing is a recent paradigm for distributed computing as a service where dynamically scalable and often virtualized resources are provided over the Internet; it typically incorporates combinations of infrastructure as a service, platform as a service, and software as a service [209]. As the paradigm is relatively new, so are the considerations of its connections to semantic technologies: Reference [210] suggests an ontological clarification of basic cloud computing concepts. As the paradigm poses high requirements to cloud service providers, internal knowledge management as well as automated, rule-based system management may be useful. First ideas in this direction have already been communicated by leading-edge companies (like http://www.fluidops.com/). It should also be noted that besides the potential benefits of semantic technologies for cloud computing, it is also a thrilling question, which potentials cloud computing could offer to the Semantic Web [211].

Regarding the state of practice in Semantic Middleware, Semantic Web Services (SWS) are emphasized here, and the reader is pointed to a set of literature about SWS application prototypes [17, 186, 212]. Nevertheless, SWS are not yet as far in the practical, commercial take-up as, for instance, semantic search (see also [213, 214]) is. More convincing application domains seem to be those where serious information-integration challenges have already been a problem for a long time, where standardization and metadata-based approaches are already well-known, and where Web Services are already a widespread technology – examples comprise geospatial information provision [215, 216], health-care services [217, 218], and eGovernment [219, 220].

13.3.2.6 Semantic Software Engineering

Closely related to the idea of semantic middleware is the idea of Semantic Software Engineering, which is thus far not so deeply investigated or well-developed. By definition, the unique tasks of Software Engineering are more about software-development time than about runtime. But also there, the same questions arise, for example, *searching* for software components to be reused, thinking about what changes to make for *integrating* with other components, etc. However, in contrast to completely open and dynamic Web scenarios, several challenges for Semantic Web approaches (like standards enforcement, incentives, trust, security, provenance) may be easier to address in the "controlled environment" of organization-internal software development. The potential role of ontologies and semantic technologies in SE is surveyed in [221, 222]. The classification from [223] is depicted in *Fig. 13.9*:

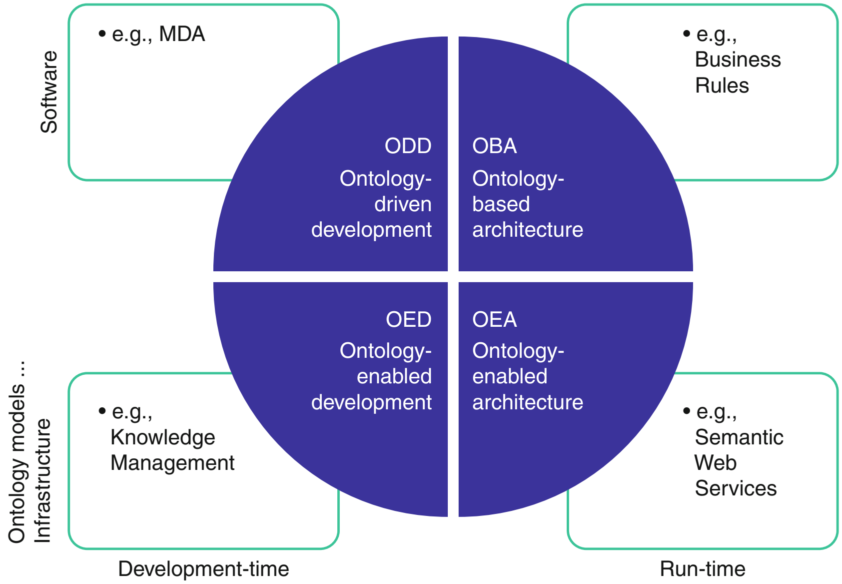

Fig. 13.9
Ontologies in Software Engineering (according to Happel and Seedorf)

Ontology-driven development (ODD) is about using ontologies at development time for describing the problem domain of the software to be developed. Prime examples are the approaches in the context of model-driven development (MDD) [224]. Benefits from using ontologies comprise reduced language ambiguity and automated validation and consistency checking.

Ontology-enabled development (OED) uses ontologies at development time for supporting software engineers with their tasks. For example, for the *identification of software components as reuse candidates* (component search), practically all aspects of semantic search can come into play (see [225] for an example). Reference [226] describes the Open Source software-development process with three ontologies, about (1) code, (2) bugs, and (3) interactions between members of the developer community. Such interactions are centered around artifacts, which may automatically be cross-linked through metadata; this supports better finding and the proactive recommendation of interesting information, for example, for bug resolution. In the same spirit, recent developments comprise, for example, the **baetle** ontology for software bugs and bug tracking systems as well as several endeavors in semantic search for bug tracking information on the Web [134], or the **EvoOnt** software-evolution OWL ontology which – together with the **iSPARQL** similarity-aware ontology query language – can solve several software analysis tasks, such as the assessment of the amount of change between releases, the computation of software design metrics, or the detection of "code smells." Reference [227] aggregates manifold feeds published by the different tools of a software forge. To this end, collected data are semantically reformatted into

RDF, Dublin Core, DOAP, and FOAF. The resulting semantic data can then be processed, republished, or displayed to project members in order to visualize and analyze a live picture of community activities in Open Source software development.

Ontology-based architectures (OBA) use an ontology as a primary artifact at runtime. The ontology makes up a central part of the application logic. Business rule approaches are an example for this kind of application, or rule-based systems in general.

Ontology-enabled architectures (OEA) leverage ontologies to provide infrastructure support at the runtime of a software system. This is the case for all semantic middleware approaches listed above.

13.3.3 Selected Application Domains

After this functionality-centered description, some examples are provided for application domains with many ontology-based Semantic Web applications.

13.3.3.1 Ontologies About Cultural Heritage

The cultural heritage area is a domain with a huge amount of publicly available data and information, interested in being found, such that there are already long-standing metadata standardization efforts (see also Semantic Web Services). It provides representational challenges, as almost all statements have some time and spatial aspect, which is not easy to represent and reason about. In [150], an overview of *semantic portals for cultural heritage* is given, including important vocabularies and ontologies in the area, examples for logical rules in cultural heritage KBs, and typical services of cultural heritage portals.

In this area, **CIDOC CRM** (Conceptual Reference Model) [228] is an ontology for terminology and information sharing in the cultural heritage area. As standard ISO 21127:2006, its scope is defined as the exchange and integration of heterogeneous scientific documentation relating to museum collections. The idea is that knowledge sharing between archives, libraries, and museums is facilitated. As a comprehensive metadata standard including the necessary top-level concepts for representing concrete historic events, places, people, etc., CIDOC CRM contains many concepts and complex relationships for, at the same time, representing aspects of a top-level ontology (e.g., regarding temporal reasoning), of an information ontology (describing complex metadata for artifacts), and of a domain ontology for the domain of historic and archeological and artistic artifacts (e.g., materials). There is an RDFS and an OWL version of CIDOC CRM.

In the **SCULPTEUR** project [229], existing database systems of several museums were mapped to CRM in order to allow cross-collection searching. A graph-based visualization of a simplified CRM model was the basis for browsing the metadata space with a concept browser tool. In order to provide a usable access to the huge space of instance data, the graph-based concept browser was accompanied by the mSpace multifaceted search approach (http://www.mspace.fm/). Besides the ontology-based search, SCULPTEUR also offers a shape- and color-based similarity search on 2D images or 3D objects. The **REACH** project [230] also implements a hybrid ontology- and visual-based search in cultural heritage multimedia libraries. As above, annotation metadata are mapped to CIDOC CRM, and a hybrid search algorithm can combine evidence from similar visual characteristics and similar metadata. Reference [151] addresses two typical aspects of cultural heritage portals, namely, ontology-based spatiotemporal search as well as the visualization of concepts, but focus on the fact that many historical facts change over time (population of a town, borders of a country, names of a person, etc.) – they derive explicit time series of temporal part-of ontologies and visualize change over time. Interesting from the technological viewpoint is also [231]: The authors employ OWL ontologies and forward-chaining SWRL rules in a real-time ubiquitous application, which dynamically and in a context-aware manner delivers content to audio museum guides for people who walk through a museum exhibition. The selection also takes into account psychoacoustic properties of sound objects.

13.3.3.2 Ontologies in eGovernment

eGovernment is a demanding, as well as promising application area for semantic technologies [232, 233], be it for discovering, composing, and reconfiguring eGovernment Web Services [234, 235] and enabling semantic interoperability between public administrations' software [236], for knowledge management within public administrations [237], or for eGovernment information or service portals for citizen or companies [238].

OE-gov is an initiative of TopQuadrant Inc. to collect and distribute eGovernment OWL ontologies (http://www.oegov.us/). The importance of semantic interoperability in eGovernment has also been recognized by the European Union (http://www.semic.eu/), but did not yet lead to a widespread harmonization of ontologies and semantic resources. One of the earliest topics to be expected for standardization may be life-event ontologies, which are typically used for navigating through eGovernment service portals [239]. Recently, the Linked-Open-Data initiatives of the US and UK governments fueled significantly the interest in the LOD topic (see http://www.data.gov/ and http://data.gov.uk/). More details on the relationship between the Semantic Web and eGovernment can be found in eGovernment.

13.3.3.3 Ontologies in the Life Science Domain

Probably the most important (and far developed) application domain of Semantic Web and ontology-based approaches is the area of life sciences, biotechnology, medicine, and pharmaceutical research and development [240].

Certainly the most notable ontology is the Gene Ontology (GO) [241] – worked on since the late 1990s – which currently comprises three structured controlled vocabularies (ontologies) that describe gene products in terms of their associated *biological processes*, *cellular components*, and *molecular functions* in a species-independent manner. The GO project develops and maintains those ontologies, annotates gene products with respect to them, and also provides tools for maintenance and use. The Open Biomedical Ontologies (OBO) initiative [242] consortium is pursuing a strategy to overcome the problem of the proliferation of biomedical ontologies. Existing OBO ontologies, including the GO, are undergoing coordinated reform, and new ontologies are being created on the basis of an evolving set of shared principles governing ontology development. The result shall be an expanding family of ontologies designed to be interoperable and logically well-formed.

The ultimate purpose of all such efforts is demonstrated in the *BioPortal* [155], which gives the possibility for comfortably annotating all kinds of biomedical resources (such as gene expression datasets, descriptions of radiology images, clinical-trial reports, or PubMed article abstracts) with concepts from any publicly available biomedical ontology, and provides the user at query time with an integrated view on all these different resources. In 2008, BioPortal had already annotated more than 1.1 Mio elements (mostly PubMed article abstracts, but also, for example, more than 50,000 clinical-trial descriptions), on average each element annotated with 486 concepts.

Regarding *semantic integration* in life sciences, [243] go step-by-step through a real-world data integration example from biomedical research about Parkinson's Disease and illustrate with many examples how to design ontologies and how to build wrappers for a number of existing data sources from different biomedical research fields such that the whole information space can be queried jointly through SPARQL queries. References [154, 244] make a comprehensive analysis of *semantic search* in life sciences and present the **GoPubMed** and **GoWeb** system, respectively, which employ the Gene Ontology and the MeSH as background knowledge for improved question answering and document search. The **Conceptual Open Hypermedia Service** (**COHSE**) [245] provides *navigation between Web resources* in large, dynamic, and complex knowledge spaces, supported by an ontology as a conceptual model which, together with lexical labels, drives the dynamic linking of Web resources.

In the area of medical applications, one of the very first motivations for using ontologies is that of enabling message exchange between software, which uses different disease classification systems. Here, [246] discusses how the expressive power of OWL can be used to describe transformations between different encodings.

ASMER [247] is a deployed and operational system for active, semantic, *electronic medical records*. Electronic medical records allow one to have all patient data for one person integrated at one's fingertips. *Active semantic documents* are document collections automatically annotated with regard to one or more formal ontologies, including rules working on the semantic annotations and relationships for automatic and dynamic validation and decision-making support. In this respect, they combine aspects of *semantic information integration* and of *intelligent advisory systems*. In ASMER, medical information is annotated with regard to a comprehensive OWL ontology comprising aspects such as drugs and drug interactions, indications for drugs, medical conditions, treatments, diagnoses, and procedures. RDQL rules are then used, for example, for drug-interaction checks, for drug-formulary checks, for drug-dosage range checks, for drug-allergy interaction checks, etc. **HealthFinland** as an example for a *semantic portal* in the medical area has already been sketched above.

13.4 Related Resources

The following are key references for ontologies in Computer Science.

Handbook on Ontologies (*2nd edition*) [248] – The handbook on ontologies gives an extensive overview on ontologies as a topic in Computer Science. It covers foundations of ontology languages, engineering methods, and management infrastructure, and also addresses aspects of the application and usage of ontologies. Its content is strongly influenced by the field of the Semantic Web and, thus, provides beneficial further reading on the topics covered in this chapter. In its second edition, it has been updated with recent developments in technology and augmented by many evolving application areas of ontologies.

Foundations of Semantic Web Technologies [249] – A comprehensive textbook on the foundations of Semantic Web ontology languages. It provides a good basis for University courses on Semantic Web technology with a focus on the W3C-standardized ontology languages RDFS and OWL with their underlying formal logical semantics, including exercises and solutions. Besides the technical basics on ontology languages with their syntax and semantics, the book addresses aspects of reasoning with ontologies, ontology engineering, rule extensions and querying, as well as some applications.

The Description Logic Handbook (*2nd Edition*) [11] – A comprehensive overview on Description Logics (DLs) for symbolic knowledge representation. The book covers the foundations of various DLs with their formal, model-theoretic semantics, methods of automated reasoning in DLs, and various other topics.

Semantic Web for the Working Ontologist [250] – A recent introduction to modeling Semantic Web ontologies by means of the W3C standard languages RDFS and OWL. It conveys the principles of the Semantic Web, covers the essentials of the W3C Semantic Web ontology language stack, and addresses many methodological and practical aspects of ontology engineering. It is a good reference for an in-depth course in modeling with RDFS and OWL.

Ontology Learning and Population: Bridging the Gap between Text and Knowledge [31] – The book provides a survey on ontology learning techniques. It discusses ontologies for the Semantic Web, knowledge management, information retrieval, text clustering and classification, as well as natural language processing, all in the context of ontology learning.

Online resources: Besides literature, some computational resources should also be mentioned: There is a growing number of *ontology repositories* [251] with numerous reusable ontologies that is going to reduce the cold-start problem for ontology construction, for instance:

- **OntoSelect** with currently 1,530 ontologies in RDFS, DAML, and OWL [252]
- The US National Center for Biomedical Ontology's **BioPortal** [253] with 162 ontologies containing ca. 700,000 concepts
- The European Bioinformatics Institute's **Ontology Lookup Service OLS** hosting 69 ontologies with more than 880,000 terms
- The Australian **Pronto** repository with about 230 OWL, RDF, and OBO ontologies
- The **National Finnish Ontology Service ONKI**

- **SchemaWeb**, a directory of 240 RDF schemas expressed in the RDFS, OWL, and DAML+OIL schema languages

From the technological point of view, easier, reuse-based ontology engineering must be accompanied by a better understanding of *networked ontologies* as investigated in the NeOn project; NeOn also provided extensive sample ontologies and applications in the pharmaceutical and in the agricultural domains. Results from NeOn and other projects were made accessible through the Watson "Semantic Web Gateway" (http://watson.kmi.open.ac.uk/) that mainly provides software components and Application Programming Interfaces (APIs) for exploiting the Semantic Web knowledge. See Semantic Web Search Engines for more about Watson.

A recent idea for applying semantic technologies and foster their proliferation, as well, is the approach of *Linked Open Services* (http://www.linkedopenservices.org/): exposing services, that is functionalities, on the Web using the same technologies that are associated with linked data, in particular HTTP, RDF, and SPARQL. See Semantic Web Services for more on the relationship between linked data and Web Services.

Finally, one can get a good impression of up-to-date Semantic Web applications, from:

- The World Wide Web Consortium's Best Practices and Deployment Working Group: http://www.w3.org/2001/sw/BestPractices/
- The Semantic Web Challenge: http://challenge.semanticweb.org/ and
- The Semantic Web Service Challenge: http://sws-challenge.org/

13.5 Future Issues

This chapter presented ontologies as one of the major cornerstones of Semantic Web technology. The essential characteristics of ontologies in Computer Science were explained, based on a formal ontology model, which is able to express all typical ontological modeling constructs, while being independent from a concrete ontology language. Ontology-engineering methods and tools were surveyed, elaborating on tools and methods for manual ontology construction on the one hand, and tools and methods for ontology learning from text, on the other.

A number of basic benefits and generic functionalities of ontologies were listed, and typical ontology use cases were discussed, such as semantic search and semantic portals, semantic information integration and linked open data, intelligent advisory systems, as well as semantic middleware and software engineering. Many concrete examples for Semantic Web ontologies in important domains (like life sciences or cultural heritage) and their usage in one of these use-case categories were given. Many usage examples were given; nevertheless, there are still important application domains (e.g., eLearning, eScience [254], or geospatial and environmental information services) and use cases (e.g., *patent search* [255]) that were left out.

In general, there is a broad range of knowledge structures, which can be subsumed under the ontology label; consequently, also the applications differ, depending on the question of which specific aspects of ontologies are exploited and whether the focus is more on heavyweight or on lightweight knowledge models. There is still scope for applied Computer Science to better understand which kind of ontology is a worthwhile investment in which kind of application. Already [256] pointed out the trade-offs between degree of formality, sharing scope, and stability of knowledge structures – which make certain kinds of ontologies more or less profitable. Reference [257] gives a more thorough analysis from an economic point of view and investigates not only technical, but also social, economic, and legal difficulties that constrain the space of practically possible ontologies. Some promising developments are seen toward overcoming such difficulties. A better understanding of where to use heavyweight and where to use lightweight ontologies (and how to use both together) is one such development; the advent of social efforts for ontology building and maintenance is another one – an interesting approach has been sketched above, the OBO initiative for the controlled evolution of ontologies in the community of bioinformatics researchers. The *knowledge maturing theory* of the **MATURE** project with its tools and methods for collaborative, usage-embedded ontology development is another one (e.g., the **SOBOLEO** tool [258]).

There is also a recognizable trend toward ontology reuse and leveraging structured resources in ontology development, which will grow stronger as more and more ontologies and RDF repositories become available. This way of bootstrapping Semantic Web content creation could significantly speed up the formalization of metadata on the Web and foster semantic interoperability between vocabularies.

The growing amount of semantic metadata in the Web as well as the growing "Web of Data" boosted by the LOD initiative fueled a renewed and increased interest in Semantic Search and Semantic Web Search Engines. Taking into account the scalability requirements of web-scale solutions as well as other, related particularities (e.g., imprecise queries or inconsistent data sources), systems like **Semplore** [259] reconsider many of the issues already discussed in Sect. 3.2.1. Regarding "query disambiguation," for instance, [260] investigates how informal, keyword-like user queries can be transformed into query-graphs that easily lead to SPARQL queries. Later processing steps comprise aspects like "query decomposition and planning," which were presented above in Sect. 3.2.2. Regarding post-processing of answers, semantics-aware ranking of results are being developed. The sheer mass of data also makes necessary completely new work directions (of course, not so new in the areas like databases or Internet search engines) such as continuous, efficient index update in a highly dynamic world, or top-k retrieval for delivering ranked results.

Regarding longer-term prospects, it is expected that the coming years will see a new wealth of applications coming from the fact that real world and virtual worlds are increasingly interwoven. In the area of ambient intelligence, assisted living, etc., *context-aware*, *pervasive computing* will open up new opportunities [261] for intelligent software, based on semantically rich notions of context; combining this with the approach of Semantic Web Services, *service-oriented context-aware middleware* may become a next software paradigm [262] – where semantic technologies are needed to deal with the complexity of such hardware–software landscapes, and to inject common-sense intelligence into the processing algorithms. For instance, the **openAAL** initiative suggests an Open Source context-aware middleware for Ambient Assisted Living: http://openaal.org/. The recent notion of a *Semantic Sensor Web* [263, 264] goes in the same direction and further emphasizes the need for common-sense knowledge on one hand, a great variety of potential applications on the other, and also completely new requirements with regard to scalability and efficiency of reasoning.

13.6 Cross-References

- Knowledge Management in Large Organizations
- KR and Reasoning on the Semantic Web: OWL
- Semantic Web Architecture

Acknowledgments

This work has been partially financed by the German Federal State of Baden-Württemberg, by the Federal German Ministry of Economics and Technology under the research program THESEUS ("New Technologies for the Internet of Services"), by the Deutsche Forschungsgemeinschaft (DFG) under the project Multipla ("Multi Ontology Learning Crossing the Boundaries of Domains and Languages"), and by the European Commission through projects NeOn ("Lifecycle Support for Networked Ontologies") and SEALS ("Semantic Evaluation at Large Scale"). We also acknowledge financial support for Johanna Völker through a Margarete-von-Wrangell scholarship awarded by the European Social Fund (ESF) and the Ministry of Science, Research, and the Arts Baden-Württemberg.

References

1. Guarino, N., Giaretta, P.: Ontologies and knowledge bases: towards a terminological clarification. In: Mars, N.J.I. (ed.) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, pp. 25–32. IOS Press, Amsterdam (1995)
2. Craig, E.: Ontology. In: Craig, E. (ed.) Routledge Encyclopedia of Philosophy, pp. 117–118. Routledge, New York (1998)
3. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: principles and methods. IEEE Trans. Knowl. Data Eng. **25**(1–2), 161–197 (1998)
4. Guarino, N.: Formal ontology and information systems, preface. In: Guarino, N. (ed.) Proceedings of the First International Conference on Formal Ontologies in Information Systems (FOIS 1998), Trento, pp. 3–15. IOS Press, Amsterdam (1998)
5. Grimm, S.: A unifying formal ontology model. In: Proceedings of the International Conference on Knowledge Engineering and Ontology Design (KEOD 2009), Funchal (2009)
6. Sowa, J.F.: Knowledge Representation. Brooks Cole, Pacific Grove (2000)
7. Russel, S., Norvig, P.: Artificial Intelligence – A Modern Approach. Prentice-Hall, New Jersey (1995)
8. Klyne, G., Carroll, J.: RDF concepts and abstract syntax. http://www.w3.org/TR/rdfprimer/ (2004). Accessed 4 Aug 2010
9. Brickley, D., Guha, R.V.: RDF vocabulary description language – RDF schema. http://www.w3.org/TR/rdf-schema/ (2004)

10. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: the making of a web ontology language. J. Web Semant. **1**(1), 7–26 (2003)
11. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook. Cambridge University Press, Cambridge (2003)
12. Kifer, M., Lausen, G., Wu, J.: Logical foundations of object-oriented and frame-based languages. J. ACM **42**(4), 741–843 (1995)
13. Ullman, J.D.: Principles of Database and Knowledge-Base Systems, vol. I/II. Computer Science Press, New York (1989)
14. Motik, B.: Reasoning in description logics using resolution and deductive databases. Ph.D. thesis, Universität Karlsruhe (TH) (2006)
15. Krötzsch, M., Rudolph, S., Hitzler, P.: Description logic rules. In: Proceedings of the 18th European Conference on Artificial Intelligence (ECAI 2008), Patras, pp. 80–84. IOS Press, Amsterdam (2008)
16. de Bruijn, J., Lausen, H., Polleres, A., Fensel, D.: The web service modeling language WSML: an overview. In: Proceedings of the Third European Semantic Web Conference (ESWC 2006), Budva. Lecture Notes in Computer Science, vol. 4011, pp. 590–604. Springer, Berlin (2006)
17. Studer, R., Grimm, S., Abecker, A. (eds.): Semantic Web Services: Concepts, Technologies, and Applications. Springer, Berlin (2007)
18. Miles, A., Bechhofer, S.: SKOS simple knowledge organization system reference. http://www.w3.org/TR/skos-reference (2008). Accessed 4 Aug 2010
19. Fellbaum, C.D.: WordNet – An Electronic Lexical Database. Language, Speech, & Communication. MIT Press, Cambridge (1998)
20. Golder, S., Huberman, B.A.: Usage patterns of collaborative tagging systems. J. Inf. Sci. **32**(2), 198–208 (2006)
21. Oltramari, A., Gangemi, A., Guarino, N., Masolo, C.: Sweetening ontologies with DOLCE. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2002), Siguenza. Lecture Notes in Computer Science, vol. 2473, pp. 223–233. Springer, Heidelberg (2002)
22. Niles, I., Pease, A.: Towards a standard upper ontology. In: Welty, C., Smith, B. (eds.) Proceedings of the Second International Conference on Formal Ontology in Information Systems (FOIS 2001), Ogunquit (2001)
23. Hepp, M.: Products and services ontologies: a methodology for deriving OWL ontologies from Industrial Categorization Standards. Int. J. Semant. Web Inf. Syst. **2**(1), 72–99 (2006)
24. Hepp, M.: GoodRelations: an ontology for describing products and services offers on the web. In: Gangemi, A., Euzenat, J. (eds.) Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008), Acitrezza. Lecture Notes in Computer Science, vol. 5268, pp. 329–346. Springer, Heidelberg (2008)
25. Roman, D., Keller, U., Lausen, H., Lara, R., de Bruijn, J., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D.: Web service modeling ontology. J. Appl. Ontol. **1**(1), 77–106 (2005)
26. Bojars, U., Heitmann, B., Oren, E.: A prototype to explore content and context on social community sites. In: SABRE Conference on Social Semantic Web (CSSW 2007), Leipzig (2007)
27. Guo, Y., Pan, Z., Heflin, J.: An evaluation of knowledge base systems for large OWL datasets. In: Plexousakis, D., McIlraith, S., van Harmelen, F. (eds.) Proceedings of the Third International Semantic Web Conference (ISWC 2004), Hiroshima. Lecture Notes in Computer Science, vol. 3298, pp. 274–288. Springer, Heidelberg (2004)
28. Ma, L., Yang, Y., Qiu, Z., Tong Xie, G., Pan, Y., Liu, S.: Towards a complete OWL ontology benchmark. In: Proceedings of the Third European Semantic Web Conference (ESWC 2006), Budva. Lecture Notes in Computer Science, vol. 4011, pp. 125–139. Springer, Heidelberg (2006)
29. Mizoguchi, R., Kozaki, K.: Ontology engineering environments. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks on Information Systems, 2nd edn, pp. 315–336. Springer, Heidelberg (2009)
30. Sure, Y., Staab, S., Studer, R.: Ontology engineering methodology. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks on Information Systems, 2nd edn, pp. 135–152. Springer, Heidelberg (2009)

31. Buitelaar, P., Cimiano, P. (eds.): Ontology Learning and Population: Bridging the Gap between Text and Knowledge. Frontiers in Artificial Intelligence and Applications, vol. 167. IOS Press, Amsterdam (2008)
32. Jaimes, A., Smith, J.R.: Semi-automatic, data-driven construction of multimedia ontologies. In: Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2003), Baltimore, pp. 781–784. IEEE Computer Society, Washington, DC (2003)
33. Newbold, N., Vrusias, B., Gillam, L.: Lexical ontology extraction using terminology analysis: automating video annotation. In: European Language Resources Association (ELRA) (ed.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech (2008)
34. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Discovering shared conceptualizations in Folksonomies. J. Web Semant. **6**(1), 38–53 (2008)
35. Gómez-Pérez, A., Corcho-García, O., Fernández-López, M.: Ontological Engineering. Springer, New York (2003)
36. Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C.: OWL pizzas: practical experience of teaching OWL DL – common errors & common patterns. In: Motta, E., Shadbolt, N., Stutt, A., Gibbins, N. (eds.) Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004), Whittlebury Hall. Lecture Notes in Computer Science, vol. 3257, pp. 63–81. Springer, Heidelberg (2004)
37. Day-Richter, J., Harris, M.A., Haendel, M., Lewis, S.: OBO-edit – an ontology editor for biologists. Bioinformatics **23**(16), 2198–2200 (2007)
38. Brockmans, S., Haase, P., Hitzler, P., Studer, R.: A metamodel and UML profile for rule-extended OWL DL ontologies. In: Sure, Y., Domingue, J. (eds.) The Semantic Web: Research and Applications. Lecture Notes in Computer Science, vol. 4011, pp. 303–316. Springer, Berlin (2006)
39. Bernstein, A., Kaufmann, E.: GINO – a guided input natural language ontology editor. In: Cruz, I.F., et al. (eds.) Proceedings of the Fifth International Semantic Web Conference (ISWC 2006), Athens, GA. Lecture Notes in Computer Science, vol. 4273, pp. 144–157. Springer, Heidelberg (2006)
40. Davis, B., Iqbal, A.A., Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., Handschuh, S.: RoundTrip ontology authoring. In: Proceedings of the Seventh International Semantic Web Conference (ISWC 2008), Karlsruhe. Lecture Notes in Computer Science, vol. 5318, pp. 50–65. Springer, Berlin (2008)
41. Fuchs, N.E., Kaljurand, K., Schneider, G.: Attempto controlled English meets the challenges of knowledge representation, reasoning, interoperability and user interfaces. In: Sutcliffe, G.C.J., Goebel, R.G. (eds.) Proceedings of the 19th Florida Artificial Intelligence Research Society Conference (FLAIRS 2006), Melbourne Beach, pp. 664–669. AAAI Press, Menlo Park (2006)
42. Hart, G., Johnson, M., Dolbear, C.: Rabbit: developing a control natural language for authoring ontologies. In: Proceedings of the Fifth European Semantic Web Conference (ESWC 2008), Tenerife. Lecture Notes in Computer Science, vol. 5021, pp. 348–360. Springer, Berlin (2008)
43. Krötzsch, M., Vrandecic, D., Völkel, M., Haller, H., Studer, R.: Semantic wikipedia. J. Web Semant. **5**, 251–261 (2007)
44. Tudorache, T., Noy, N., Tu, S., Musen, M.A.: Supporting collaborative ontology development in protégé. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T.W., Thirunarayan, K. (eds.) Proceedings of the Seventh International Semantic Web Conference (ISWC 2008), Karlsruhe. Lecture Notes in Computer Science, vol. 5318, pp. 17–32. Springer, Berlin (2008)
45. Wennerberg, P., Zillner, S., Möller, M., Buitelaar, P., Sintek, M.: KEMM: a knowledge engineering methodology in the medical domain. In: Eschenbach, C., Grüninger, M. (eds.) Proceedings of the Fifth International Conference on Formal Ontology in Information Systems (FOIS 2008), Saarbrücken. Frontiers in Artificial Intelligence and Applications, Saarbrucken, vol. 183, pp. 79–91. IOS Press, Amsterdam (2008)
46. Guarino, N., Welty, C.: An overview of OntoClean. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks

on Information Systems, 2nd edn, pp. 245–267. Springer, Heidelberg (2009)
47. Simperl, E., Tempich, C.: Exploring the economical aspects of ontological engineering. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks on Information Systems, 2nd edn, pp. 337–358. Springer, Heidelberg (2009)
48. Tempich, C., Simperl, E., Luczak, M., Studer, R., Pinto, H.S.: Argumentation-based ontology engineering. IEEE Intell. Syst. **22**(6), 52–59 (2007)
49. Vrandečić, D.: Ontology evaluation. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks on Information Systems, 2nd edn, pp. 293–313. Springer, Heidelberg (2009)
50. Gangemi, A., Presutti, V.: Ontology design patterns. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks on Information Systems, 2nd edn, pp. 221–243. Springer, Heidelberg (2009)
51. Presutti, V., Gangemi, A.: Content ontology design patterns as practical building blocks for web ontologies. In: Proceedings of the 27th International Conference on Conceptual Modeling (ER 2008), Barcelona, pp. 128–141. Springer, Berlin (2008)
52. Scharffe, F., Fensel, D.: Correspondence patterns for ontology alignment. In: Gangemi, A., Euzenat, J. (eds.) Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2008), Acitrezza. Lecture Notes in Computer Science, vol. 5268, pp. 83–92. Springer, Heidelburg (2008)
53. Fernández-López, M., Gómez-Pérez, A., Sierra, J.P., Sierra, A.P.: Building a chemical ontology using methontology and the ontology design environment. IEEE Intell. Syst. **14**(1), 37–46 (1999)
54. Tempich, C., Pinto, H.S., Sure, Y., Staab, S.: An argumentation ontology for DIstributed, Loosely-controlled and evolvInG Engineering processes of oNTologies (DILIGENT). In: Gómez-Pérez, A., Euzenat, J. (eds.) Proceedings of the Second European Semantic Web Conference (ESWC 2005), Heraklion. Lecture Notes in Computer Science, vol. 3532, pp. 241–256. Springer, Berlin (2005)
55. Noy, N., McGuinness, D.L.: Ontology development 101: a guide to creating your first ontology. Technical report KSL-01-05 and SMI-2001-0880, Stanford Knowledge Systems Laboratory and Stanford Medical Informatics (2001)
56. Sure, Y., Studer, R.: On-to-knowledge methodology. In: Davies, J., Fensel, D., van Harmelen, F. (eds.) On-To-Knowledge: Semantic Web enabled Knowledge Management, Chapter. 3, pp. 33–46. Wiley, New York (2002)
57. De Nicola, A., Missikoff, M., Navigli, R.: A software engineering approach to ontology building. Inf. Syst. **34**(2), 258–275 (2009)
58. Kotis, K., Vouros, A.: Human-centered ontology engineering: the HCOME methodology. Knowl. Inf. Syst. **10**(1), 109–131 (2006)
59. Suárez-Figueroa, M.C., et al.: D5.4.1 NeOn methodology for building contextualized ontology networks. Technical report 5.4.1, NeOn deliverable (2009)
60. Jarrar, M., Meersman, R.: Ontology engineering – the DOGMA approach. In: Dillon, T.S., Chang, E., Meersman, R., Sycara, K. (eds.) Advances in Web Semantics I. Lecture Notes in Computer Science, vol. 4891. Springer, Berlin (2008)
61. Grüninger, M., Fox, M.: Methodology for the design and evaluation of ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing at International Joint Conference on Artificial Intelligence (IJCAI 1995), Montreal (1995)
62. Uschold, M., King, M.: Towards a methodology for building ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing at International Joint Conference on Artificial Intelligence (IJCAI 1995), Montreal (1995)
63. Garcia Castro, A., Rocca-Serra, P., Stevens, R., Taylor, C., Nashar, K., Ragan, M.A., Sansone, S.-A.: The use of concept maps during knowledge elicitation in ontology development processes-the nutrigenomics use case. BMC Bioinform. **7**, 267 (2006)
64. Noy, N., Fergerson, R., Musen, M.: The knowledge model of protégé-2000: combining interoperability and flexibility. In: Dieng, R., Corby, O. (eds.) Proceedings of the 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2000), Juom-les-pins. Lecture Notes in Artificial Intelligence, vol. 1937, pp. 17–32. Springer, Berlin (2000)

65. Arpírez, J.C., Corcho, O., Fernández-López, M., Gómez-Pérez, A.: WebODE: a scalable workbench for ontological engineering. In: Proceedings of the First International Conference on Knowledge Capture (K-CAP 2001), Victoria, pp. 6–13. ACM, New York (2001)
66. Sure, Y., Angele, J., Staab, S.: OntoEdit: multifaceted inferencing for ontology engineering. J. Data Semant. Lecture Notes in Computer Science, **2800**, 128–152 (2003)
67. Völker, J., Vrandečić, D., Sure, Y., Hotho, A.: AEON – an approach to the automatic evaluation of ontologies. J. Appl. Ontol. **3** (1–2), 41–62, ISSN 1570-5838. IOS Press, Amsterdam (2008)
68. Cimiano, P., Mädche, A., Staab, S., Völker, J.: Ontology learning. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. International Handbooks on Information Systems, 2nd edn, pp. 245–267. Springer, Heidelberg (2009)
69. Grimnes, G.A., Edwards, P., Preece, A.D.: Learning meta-descriptions of the FOAF network. In: McIlraith, S.A., et al. (eds.) Proceedings of the Third International Semantic Web Conference (ISWC 2004), Hiroshima. Lecture Notes in Computer Science, vol. 3298, pp. 152–165. Springer, Berlin (2004)
70. Basili, R., Moschitti, A., Pazienza, M.T., Zanzotto, F.M.: A contrastive approach to term extraction. In: Proceedings of the Fourth Terminology and Artificial Intelligence Conference (TIA 2001), Nancy, pp. 119–128 (2001)
71. Drouin, P.: Detection of domain specific terminology using corpora comparison. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004), pp. 79–82. European Language Resources Association, Lisbon (2004)
72. Kilgarriff, A.: Comparing corpora. Int. J. Corpus Linguist. **1**(6), 97–103 (2001)
73. Brunzel, M.: The XTREEM methods for ontology learning from web documents. In: Buitelaar, P., Cimiano, P. (eds.) Ontology Learning and Population: Bridging the Gap between Text and Knowledge. Frontiers in Artificial Intelligence and Applications, vol. 167, pp. 3–26. IOS Press, Amsterdam (2008)
74. Jacquemin, C.: Spotting and Discovering Terms Through Natural Language Processing. MIT Press, Cambridge (2001)
75. Witschel, H.F.: Terminology extraction and automatic indexing – comparison and qualitative evaluation of methods. In: Madsen, B.N., Thomsen, H.E. (eds.) Proceedings of the Seventh International Conference on Terminology and Knowledge Engineering (TKE 2005). Association for Terminology and Knowledge Transfer, Copenhagen (2005)
76. Harris, Z.S.: Distributional structure. Word **10**(23), 146–162 (1954)
77. Pantel, P., Crestan, E., Borkovsky, A., Popescu, A.-M., Vyas, V.: Web-scale distributional similarity and entity set expansion. In: Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP 2009), Singapore, pp. 938–947 (2009)
78. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th International Conference on Computational Linguistics (COLING 1992), Nantes, pp. 539–545 (1992)
79. Ogata, N., Collier, N.: Ontology express: statistical and non-monotonic learning of domain ontologies from text. In: Proceedings of the Workshop on Ontology Learning and Population (OLP 2004) at the European Conference on Artificial Intelligence (ECAI 2004), Valencia (2004)
80. Cimiano, P., Handschuh, S., Staab, S.: Towards the self-annotating web. In: Proceedings of the 13th International World Wide Web Conference (WWW 2004), Manhattan, pp. 462–471. ACM, New York (2004)
81. Potrich, A., Pianta, E.: L-ISA: learning domain specific isa-relations from the web. In: European Language Resources Association (ed.) Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008), Marrakech (2008)
82. Blohm, S., Cimiano, P.: Using the web to reduce data sparseness in pattern-based information extraction. In: Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2007), Bled, pp. 18–29. Springer, Heidelberg (2007)
83. Cimiano, P., Ladwig, G., Staab, S.: Gimme the context: context-driven automatic semantic annotation with C-PANKOW. In: Ellis, A., Hagino, T. (eds.) Proceedings of the 14th International World Wide Web Conference

(WWW 2005), Chiba, pp. 332–341. ACM, New York (2005)
84. Keller, F., Lapata, M., Ourioupina, O.: Using the web to overcome data sparseness. In: Hajič, J., Matsumoto, Y. (eds.) Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002), Philadelphia, pp. 230–237. Association for Computational Linguistics (2002)
85. Resnik, P., Smith, N.A.: The web as a parallel corpus. Comput. Linguist. **29**(3), 349–380 (2003)
86. Blohm, S., Cimiano, P., Stemle, E.: Harvesting relations from the web – quantifying the impact of filtering functions. In: Proceedings of the 22nd Conference on Artificial Intelligence (AAAI 2007), Vancouver, pp. 1316–1323. AAAI Press (2007)
87. Brewster, C., Iria, J., Zhang, Z., Ciravegna, F., Guthrie, L., Wilks, Y.: Dynamic iterative ontology learning. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2007), Borovets (2007)
88. Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Web-scale information extraction in KnowItAll (preliminary results). In: Proceedings of the 13th International World Wide Web Conference (WWW 2004), Manbattan, pp. 100–109. ACM, New York (2004)
89. Pantel, P., Pennacchiotti, M.: Espresso: leveraging generic patterns for automatically harvesting semantic relations. In: Proceedings of the 21st International Conference on Computational Linguistics (COLING 2006) and 44th Annual Meeting of the Association for Computational Linguistics (ACL 2006), Sydney, pp. 113–120. The Association for Computer Linguistics (2006)
90. Caraballo, S.A.: Automatic construction of a hypernym-labeled noun hierarchy from text. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL 1999), Maryland, pp. 120–126. Association for Computational Linguistics (1999)
91. Cimiano, P., Hotho, A., Staab, S.: Comparing conceptual, divise and agglomerative clustering for learning taxonomies from text. In: López de Mántaras, R., Saitta, L. (eds.) Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), Valencia, pp. 435–439. IOS Press, Amsterdam (2004)
92. Faure, D., Nédellec, C.: Knowledge acquisition of predicate argument structures from technical texts using machine learning: the system ASIUM. In: Fensel, D., Studer, R. (eds.) Proceedings of the 11th International Conference on Knowledge Engineering and Knowledge Management (EKAW 1999), Dagstubl Castle. Lecture Notes in Computer Science, vol. 1621, pp. 329–334. Springer, Berlin (1999)
93. Cimiano, P., Völker, J.: Towards large-scale, open-domain and ontology-based named entity classification. In: Angelova, G., Bontcheva, K., Mitkov, R., Nicolov, N. (eds.) Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005), Borovets, pp. 166–172. INCOMA, Bulgaria (2005)
94. Coppola, B., Gangemi, A., Gliozzo, A.M., Picca, D., Presutti, V.: Frame detection over the semantic web. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E.P.B. (eds.) Proceedings of the Sixth European Semantic Web Conference (ESWC 2009). Lecture Notes in Computer Science, vol. 5554, pp. 126–142. Springer, Berlin (2009)
95. Suchanek, F.M., Sozio, M., Weikum, G.: SOFIE: a self-organizing framework for information extraction. In: Proceedings of the 18th International Conference on World Wide Web (WWW 2009), Madrid, pp. 631–640. ACM, New York (2009)
96. Etzioni, O., Banko, M., Soderland, S., Weld, D.S.: Open information extraction from the web. Commun. ACM **51**(12), 68–74 (2008)
97. Schutz, A., Buitelaar, P.: RelExt: a tool for relation extraction from text in ontology extension. In: Gil, Y., Motta, E., Benjamins, V.R. (eds.) Proceedings of the Fourth International Semantic Web Conference (ISWC 2005), Galway. Lecture Notes in Computer Science, vol. 3729, pp. 593–606. Springer, Berlin (2005)
98. Cimiano, P., Hartung, M., Ratsch, E.: Finding the appropriate generalization level for binary relations extracted from the Genia Corpus. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation

(LREC 2006), Genoa, pp. 161–169. European Language Resources Association (2006)

99. Heyer, G., Läuter, M., Quasthoff, U., Wittig, T., Wolff, C.: Learning relations using collocations. In: Mädche, A., Staab, S., Nédellec, C., Hovy, E.H. (eds.) Proceedings of the Second Workshop on Ontology Learning (OL 2001) at International Joint Conference on Artificial Intelligence (IJCAI 2001), Seattle, vol. 38. CEUR-WS.org (2001)
100. Antunes, C.: Mining rules in the Onto4AR framework for updating domain ontologies. In: Proceedings of the IADIS European Conference on Data Mining (ECDM 2007), Lisbon, pp. 95–101. IADIS Press (2007)
101. Mädche, A., Staab, S.: Discovering conceptual relations from text. In: Horn, W. (ed.) Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000), Berlin, pp. 321–325. IOS Press, Amsterdam (2000)
102. Kavalec, M., Svátek, V.: A study on automated relation labelling in ontology learning. In: Buitelaar, P., Cimiano, P., Magnini, B. (eds.) Ontology Learning from Text: Methods, Evaluation and Applications. Frontiers in Artificial Intelligence and Applications, vol. 123, pp. 44–58. IOS Press, Amsterdam (2005)
103. Rudolph, S.: Acquiring generalized domain-range restrictions. In: Medina, R., Obiedkov, S. (eds.) Proceedings of the Sixth International Conference on Formal Concept Analysis (ICFCA 2008), Montreal. Lecture Notes in Artificial Intelligence, vol. 4933, pp. 32–45. Springer, Heidelberg (2008)
104. Völker, J., Rudolph, S.: Fostering web intelligence by semi-automatic OWL ontology refinement. In: Proceedings of the Seventh International Conference on Web Intelligence (WI 2008), Sydney. IEEE Washington, DC (2008)
105. Völker, J., Vrandečić, D., Sure, Y., Hotho, A.: Learning disjointness. In: Franconi, E., Kifer, M., May, W. (eds.) Proceedings of the Fourth European Semantic Web Conference (ESWC 2007), Innsbruck. Lecture Notes in Computer Science, vol. 4519, pp. 175–189. Springer, Berlin (2007)
106. Fanizzi, N., Iannone, L., Palmisano, I., Semeraro, G.: Concept formation in expressive description logics. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) Proceedings of the 15th European Conference on Machine Learning (ECML). Lecture Notes in Artificial Intelligence, vol. 3201. Springer, Berlin (2004)
107. Hellmann, S., Lehmann, J., Auer, S.: Learning of OWL class descriptions on very large knowledge bases. Int. J. Semant. Web Inf. Syst. **5**(2), 25–48 (2009)
108. Baader, F., Ganter, B., Sertkaya, B., Sattler, U.: Completing description logic knowledge bases using formal concept analysis. In: Veloso, M.M. (ed.) Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), Hyderabad, pp. 230–235. AAAI Press, Menlo Park (2007)
109. Völker, J., Rudolph, S.: Lexico-logical acquisition of OWL DL axioms – an integrated approach to ontology refinement. In: Medina, R., Obiedkov, S. (eds.) Proceedings of the Sixth International Conference on Formal Concept Analysis (ICFCA 2008), Montreal. Lecture Notes in Artificial Intelligence, vol. 4933, pp. 62–77. Springer, Berlin (2008)
110. Shamsfard, M., Barforoush, A.A.: Learning ontologies from natural language texts. Int. J. Hum. Comput. Stud. **60**(1), 17–63 (2004)
111. Völker, J.: Learning expressive ontologies. Studies on the semantic web, vol. 2. AKA Verlag/IOS Press, Heidelberg (2009)
112. d'Aquin, M., Motta, E., Dzbor, M., Gridinoc, L., Heath, T., Sabou, M.: Collaborative semantic authoring. IEEE Intell. Syst. **23**(3), 80–83 (2008)
113. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T.M., Nigam, K., Slattery, S.: Learning to construct knowledge bases from the world wide web. Artif. Intell. **118**(1–2), 69–113 (2000)
114. Mädche, A., Volz, R.: The text-to-onto ontology extraction and maintenance system. In: Proceedings of the Workshop on Integrating Data Mining and Knowledge Management at the First International Conference on Data Mining (ICDM 2001), San Jose (2001)
115. Buitelaar, P., Olejnik, D., Sintek, M.: A protégé plug-in for ontology extraction from text based on linguistic analysis. In: Proceedings of the First European Semantic Web Symposium (ESWS 2004), Heraklion. Lecture Notes in Computer Science, vol. 3053, pp. 31–44. Springer, Berlin (2004)

116. Morita, T., Shigeta, Y., Sugiura, N., Fukuta, N., Izumi, N., Yamaguchi, T.: DOODLE-OWL: OWL-based semi-automatic ontology development environment. In: Proceedings of the Workshop on Evaluation of Ontology-Based Tools (EON 2004) at the International Semantic Web Conference (ISWC 2004), Hiroshima (2004)
117. Cimiano, P., Völker, J.: Text2Onto – a framework for ontology learning and data-driven change discovery. In: Montoyo, A., Munoz, R., Metais, E. (eds.) Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB 2005), Alicante. Lecture Notes in Computer Science, vol. 3513, pp. 227–238. Springer, Berlin (2005)
118. Velardi, P., Navigl, R., Cucchiarelli, A., Neri, F.: Evaluation of OntoLearn, a methodology for automatic learning of domain ontologies. In: Buitelaar, P., Cimiano, P., Magnini, B. (eds.) Ontology Learning from Text: Methods, Evaluation and Applications. Frontiers in Artificial Intelligence and Applications, vol. 123. IOS Press, Amsterdam (2005)
119. Nováček, V., Smrž, P.: OLE – a new ontology learning platform. In: RANLP-Workshop on Text Mining Research, Practice and Opportunities, Borovets, pp. 12–16. Incoma (2005)
120. Fortuna, B., Grobelnik, M., Mladenic, D.: OntoGen: semi-automatic ontology editor. In: Smith, M.J., Salvendy, G. (eds.) Human Interface and the Management of Information. Interacting in Information Environments. Symposium on Human Interface held as Part of HCI International. Lecture Notes in Computer Science, vol. 4558, pp. 309–318. Springer, Berlin (2007)
121. Manzano-Macho, D., Gómez-Pérez, A., Borrajo, D.: Unsupervised and domain independent ontology learning: combining heterogeneous sources of evidence. In: European Language Resources Association (ed.) Proceedings of the Sixth International Language Resources and Evaluation (LREC 2008), Marrakech (2008)
122. Nováček, V., Laera, L., Handschuh, S., Davis, B.: Infrastructure for dynamic knowledge integration – automated biomedical ontology extension using textual resources. J. Biomed. Inf. **41**(4), 816–828 (2008)
123. Gacitua, R., Sawyera, P., Rayson, P.: A flexible framework to experiment with ontology learning techniques. Knowl.-Based Syst. **21**(3), 192–199 (2008); 27th SGAI International Conference on Artificial Intelligence (AI 2008), Cambridge (2008)
124. Sertkaya, B.: OntoComP: a protege plugin for completing OWL ontologies. In: Aroyo, L., Traverso, P. (eds.) Demo Proceedings of the Sixth European Semantic Web Conference (ESWC 2009), Heraklion. Lecture Notes in Computer Science, vol. 5554, pp. 898–902. Springer, Berlin (2009)
125. Simperl, E., Tempich, C., Vrandečić, D.: A methodology for ontology learning. In: Buitelaar, P., Cimiano, P. (eds.) Ontology Learning and Population: Bridging the Gap between Text and Knowledge. Frontiers in Artificial Intelligence and Applications, vol. 167, pp. 225–249. IOS Press, Amsterdam (2008)
126. Nováček, V., Dabrowski, M., Kruk, S.R., Handschuh, S.: Extending community ontology using automatically generated suggestions. In: Wilson, D., Sutcliffe, G. (eds.) Proceedings of the 20th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2007), Key West (2007)
127. Blomqvist, E.: OntoCase – a pattern-based ontology construction approach. In: Meersman, R., Tari, Z. (eds.) On the Move to Meaningful Internet Systems 2007: (OTM Conferences). Lecture Notes in Computer Science, vol. 4803, pp. 971–988. Springer, Berlin (2007)
128. Völker, J., Blomqvist, E.: D3.8.2 evaluation of methods for contextualized learning of networked ontologies. Technical report 3.8.2. Institute AIFB, University of Karlsruhe, NeOn deliverable (2009)
129. Dellschaft, K., Staab, S.: On how to perform a gold standard based evaluation of ontology learning. In: Cruz, I.F., et al. (eds.) Proceedings of the Fifth International Semantic Web Conference (ISWC 2006), Athens, GA. Lecture Notes in Computer Science, vol. 4273, pp. 228–241. Springer, Heidelberg (2006)
130. Brewster, C., Jupp, S., Luciano, J., Shotton, D., Stevens, R.D., Zhang, Z.: Issues in learning an ontology from text. BMC Bioinform. **10** (Suppl 5), S1 (2009)
131. Abecker, A., Bernardi, A., Hinkelmann, K., Kühn, O., Sintek, M.: Toward a technology for

organizational memories. IEEE Intell. Syst. **13**(3), 40–48 (1998)
132. Schmidt, A.: Bridging the gap between knowledge management and e-learning with context-aware corporate learning solutions. In: Althoff, K.-D., Dengel, A., Bergmann, R., Nick, M., Roth-Berghofer, T. (eds.) Proceedings of the Third Conference on Professional Knowledge Management (WM 2005), Kaiserslautern. Lecture Notes in Artificial Intelligence, vol. 3782, pp. 203–213. Springer, Berlin (2005)
133. Stojanovic, N.: Ontology-based information retrieval: methods and tools for cooperative query answering. Ph.D. thesis, Universität Karlsruhe (TH) (2005)
134. Tran, H.M., Lange, C., Chulkov, G., Schönwälder, J., Kohlhase, M.: Applying semantic techniques to search and analyze bug tracking data. J. Netw. Syst. Manag. **17**(3), 285–308 (2009)
135. Bhogal, J., Macfarlane, A., Smith, P.: A review of ontology-based query expansion. Inf. Process. Manag. **43**(4), 866–886 (2007)
136. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
137. van Rijsbergen, C.J., Crestani, F., Lalmas, M. (eds.): Information Retrieval: Uncertainty and Logics: Advanced Models for the Representation and Retrieval of Information. Kluwer, Norwell (1998)
138. Meghini, C., Sebastiani, F., Straccia, U., Thanos, C.: A model of information retrieval based on a terminological logic. In: Korfhage, R., Rasmussen, E.M., Willett, P. (eds.) SIGIR, pp. 298–307. ACM, New York (1993)
139. Richter, M.M., Aamodt, A.: Case-based reasoning foundations. Knowl. Eng. Rev. **20**(3), 203–207 (2005)
140. Lei, Y., Motta, E., Domingue, J.: OntoWeaver-S: supporting the design of knowledge portals. In: Motta, E., Shadbolt, N., Stutt, A., Gibbins, N. (eds.) Proceedings of the 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004), Whittlebury Hall. Lecture Notes in Computer Science, vol. 3257, pp. 216–230. Springer, Berlin (2004)
141. Abel, F., Henze, N.: User awareness and personalization in semantic portals. In: Proceedings of the Fourth Internationl Semantic Web Conference (ISWC 2005), Galway. Lecture Notes in Computer Science, vol. 3729. Springer, Berlin (2005)
142. Mäkelä, E., Hyvönen, E., Saarela, S., Viljanen, K.: OntoViews – a tool for creating semantic web portals. In: McIlraith, S.A., et al. (eds.) Proceedings of the Third International Semantic Web Conference (ISWC 2004), Hiroshima. Lecture Notes in Computer Science, vol. 3298, pp. 797–811. Springer (2004)
143. Maedche, A., Staab, S., Stojanovic, N., Studer, R., Sure, Y.: Semantic portal: the SEAL approach. In: Fensel, D., Hendler, J.A., Lieberman, H., Wahlster, W. (eds.) Spinning the Semantic Web, pp. 317–359. MIT Press, Cambridge (2003)
144. Lara, R., Han, S.H., Lausen, H., Stollberg, M., Ding, Y., Fensel, D.: An evaluation of semantic web portals. In: Proceedings of the IADIS Applied Computing International Conference, Lisbon (2004)
145. Hildebrand, M., van Ossenbruggen, J.R., Hardman, L.: An analysis of search-based user interaction on the semantic web. Technical report. Centrum Wiskunde & Informatica, Amsterdam. http://nwww.cwi.nl/en/system/files/u9/INS-E0706.pdf (2007). Accessed 4 Aug 2010
146. Huu Hoang, H., Min Tjoa, A.: The state of the art of ontology-based query systems: a comparison of existing approaches. In: Proceedings of the IEEE International Conference on Computing and Informatics (ICOCI 2006), Kuala Lumpur (2006)
147. Mangold, C.: A survey and classification of semantic search approaches. Int. J. Metadata Semant. Ontol. **2**(1), 23–34 (2007)
148. Bhagdev, R., Butters, J., Chakravarthy, A., Chapman, S., Dadzie, A.-S., Greenwood, M.A., Iria, J., Ciravegna, F.: Doris: managing document-based knowledge in large organisations via semantic web technologies. In: Golbeck, J., Mika, P. (eds.) Semantic Web Challenge, CEUR Workshop Proceedings, vol. 295. CEUR-WS.org (2007)
149. Biesalski, E., Abecker, A.: Skill-profile matching with similarity measures. In: Manolopoulos, Y., Filipe, J., Constantopoulos, P., Cordeiro, J. (eds.) ICEIS (Selected Papers). Lecture Notes in Business Information Processing, vol. 3, pp. 210–218. Springer, Berlin (2006)

150. Hyvönen, E.: Semantic Portals for Cultural Heritage. Springer, Heidelberg (2009)
151. Kauppinen, T., Väätäinen, J., Hyvönen, E.: Creating and using geospatial ontology time series in a semantic cultural heritage portal. In: Bechhofer, S., et al. (eds.) Proceedings of the Fifth European Semantic Web Conference (ESWC 2008), Tenerife. Lecture Notes in Computer Science, vol. 5021, pp. 110–123. Springer, Berlin (2008)
152. Schreiber, G., Amin, A., Aroyo, L., van Assem, M., de Boer, V., Hardman, L., Hildebrand, M., Omelayenko, B., van Osenbruggen, J., Tordai, A., Wielemaker, J., Wielinga, B.: Semantic annotation and search of cultural-heritage collections: the multimediaN E-culture demonstrator. J. Web Semant. **6**(4), 243–249 (2008)
153. Wang, Y., Stash, N., Aroyo, L., Gorgels, P., Rutledge, L., Schreiber, G.: Recommendations based on semantically enriched museum collections. J. Web Semant. **6**(4), 283–290 (2008)
154. Alexopoulou, D., et al.: GoPubMed: ontology-based literature search for the life sciences. http://rewerse.net/deliverables/m42/a2-d7.pdf (2007). Accessed 1 Jan 2011
155. Jonquet, C., Musen, M.A., Shah, N.: A system for ontology-based annotation of biomedical data. In: Bairoch, A., Cohen Boulakia, S., Froidevaux, C. (eds.) Proceedings of the Fifth International Workshop on Data Integration in the Life Sciences (DILS 2008), Evry. Lecture Notes in Computer Science, vol. 5109, pp. 144–152. Springer, Heidelberg (2008)
156. Suominen, O., Hyvönen, E., Viljanen, K., Hukka, E.: HealthFinland-a national semantic publishing network and portal for health information. J. Web Semant. **7**(4), 287–297 (2009)
157. Tzouveli, P., Schmidt, A., Schneider, M., Symvonis, A., Kollias, S.: Adaptive reading assistance for the inclusion of learners with dyslexia. In: Proceedings of the IEEE Conference on Advanced Learning Technologies (ICALT 2008), Santander, Cantabria. IEEE Computer Society, Washington, DC (2008)
158. Busse, S. Kutsche, R.-D., Leser, U., Weber, H.: Federated information systems: concepts, terminology and architectures. Technical report, Fachbereich Informatik, TU Berlin. http://user.cs.tu-berlin.de/~rkutsche/Publikationen/index.html (1999). Accessed 4 Aug 2010
159. Wiederhold, G., Genesereth, M.: The conceptual basis for mediation services. IEEE Intell. Syst. **12**(5), 38–47 (1997)
160. Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., Hübner, S.: Ontology-based integration of information – a survey of existing approaches. In: Stuckenschmidt, H. (ed.) Workshop: Ontologies and Information Sharing (IJCAI 2001), Seattle (2001)
161. Cruz, I.F., Xiao, H.: Using a layered approach for interoperability on the semantic web. In: Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE 2003). IEEE Computer Society
162. Doan, A., Halevy, A.Y.: Semantic integration research in the database community: a brief survey. AI Mag. **26**(1), 83–94 (2005)
163. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, Heidelberg (2007)
164. Kalfoglou, Y., Schorlemmer, M.: Ontology mapping: the state of the art. Knowl. Eng. Rev. **18**(1), 1–31 (2003)
165. Saggion, H., Funk, A., Maynard, D., Bontcheva, K.: Ontology-based information extraction for business intelligence. In: Aberer, K., et al. (eds.) Proceedings of the Sixth International Semantic Web Conference, Second Asian Semantic Web Conference (ISWC/ASWC 2007), Busan. Lecture Notes in Computer Science, vol. 4825, pp. 843–856. Springer, Berlin (2007)
166. Kalfoglou, Y., Hu, B., Reynolds, D., Shadbolt, N.: Semantic integration technologies survey – 6th month deliverable of the project CROSI: capturing representing and operationalising semantic integration. Technical report. University of Southampton, School of Electronics and Computer Science. http://www.aktors.org/crosi/deliverables/(2005). Accessed 4 Aug 2010
167. Maier, A., Schnurr, H.-P., Sure, Y.: Ontology-based information integration in the automotive industry. In: Proceedings of the Second International Semantic Web Conference (ISWC 2003), Sanibel Island. Lecture Notes in Computer Science, vol. 2870, pp. 897–912. Springer, Berlin (2003)
168. Lu, J., Wang, S., Capretz, M.A.M.: A service oriented ontology management framework in the automotive retail domain. In: Proceedings of the 11th IEEE International Conference on

Computational Science and Engineering (CSE 2008) – Workshops, Sao Paulo, pp. 239–244 (2008)

169. Vujasinovic, M., Ivezic, N., Kulvatunyou, B., Barkmeyer, E., Missikoff, M., Taglino, F., Marjanovic, Z., Miletic, I.: A semantic-mediation architecture for interoperable supply-chain applications. Int. J. Comput. Integr. Manuf. **22**(6), 549–561 (2008)
170. Barillot, C., Benali, H., Dameron, O., Dojat, M., Gaignard, A., Gibaud, B., Kinkingnehun, S., Matsumoto, J.-P., Pelegrini-Issac, M., Simon, E., Temal, L., Valabregue, R.: Federating distributed and heterogeneous information sources in neuroimaging: the NeuroBase project. Technical report 1712. http://www.irisa.fr/visages/demo/Neurobase/Download/IRISAResearch Report 1712.pdf (2005). Accessed 4 Aug 2010
171. Angele, J., Gesmann, M.: Semantic information integration with software AGs information integrator. In: Proceedings of the Second International Conference on Rules and Rule Markup Languages for the Semantic Web (RuleML 2006), Athens, GA. http://2006.ruleml.org/ (2006). Accessed 4 Aug 2010
172. Angele, J., Gesmann, M.: Integration of customer information using the semantic web. In: Cardoso, J., et al. (eds.) The Semantic Web: Real-World Applications from Industry. Semantic Web And Beyond Computing for Human Experience, vol. 6, pp. 191–208. Springer, New York (2007)
173. Bizer, C., Heath, T., Berners-Lee, T.: Linked data – the story so far. Int. J. Semant. Web Inf. Syst. **5**(3), 1–22 (2009)
174. Coetzee, P., Heath, T., Motta, E.: SparqPlug: generating linked data from legacy HTML, SPARQL and the DOM. In: Proceedings of the First International Workshop on Linked Data on the Web (LDOW 2008) at 17th International World Wide Web Conference (WWW 2008), Beijing. http://events.linkeddata.org/ldow2008/papers/05-coetzee-heath-sparqplug.pdf (2008). Accessed 4 Aug 2010
175. Becker, C., Bizer, C.: DBpedia mobile – a location-aware semantic web client. In: Proceedings of the Semantic Web Challenge at Seventh International Semantic Web Conference (ISWC 2008), Karlsruhe. Lecture Notes in Computer Science, vol. 5318. Springer, Berlin (2008)
176. Das Sarma, A., Dong, X., Halevy, A.Y.: Bootstrapping pay-as-you-go data integration systems. In: Tsong-Li Wang, J. (ed.) Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2008), Vancouver, pp. 861–874. ACM, New York (2008)
177. Kobilarov, G., Scott, T., Raimond, Y., Oliver, S., Sizemore, C., Smethurst, M., Lee, R.: Media meets semantic web – how the BBC uses DBpedia and linked data to make connections. In: Proceedings of the Sixth European Semantic Web Conference (ESWC 2009), Heraklion. Lecture Notes in Computer Science, vol. 5554. pp. 723–737. Springer, Berlin (2009)
178. Jackson, P.: Introduction to Expert Systems, 3rd edn. Addison-Wesley, Boston (1998)
179. Payne, V.L., Metzler, D.P.: Hospital Care Watch (HCW): an ontology and rule-based intelligent patient management assistant. In: Proceedings of the 18th International Symposium on Computer-Based Medical Systems (CBMS 2005), Dublin, pp. 479–484. IEEE Computer Society, Washington, DC (2005)
180. Lee, C.-S., Wang, M.-H.: Ontology-based intelligent healthcare agent and its application to respiratory waveform recognition. Expert Syst. Appl. **33**(3), 606–619 (2007)
181. Lee, C.-S., Wang, M.-H.: Ontological fuzzy agent for electrocardiogram application. Expert Syst. Appl. **35**(3), 1223–1236 (2008)
182. Bobillo, F., Delgado, M., Gmez-Romero, J.: Representation of context-dependant knowledge in ontologies: a model and an application. Expert Syst. Appl. **35**(4), 1899–1908 (2008)
183. Angele, J., Mönch, E., Nierlich, A., Oppermann, H., Rudat, H., Schnurr, H.-P.: Applications and good practices of semantic technologies. http://www.eddie-moench.de/publications (2006). Accessed 4 Aug 2010
184. Britton, C.: IT Architectures and Middleware: Strategies for Building Large. Integrated Systems. Addison-Wesley, Boston (2001)
185. Taylor, I.J., Harrison, A.: From P2P and Grids to Services on the Web-Evolving Distributed Communities, 2nd edn. Springer, London (2009)
186. Kuropka, D., Tröger, P., Staab, S., Weske, M. (eds.): Semantic Service Provisioning. Springer, Berlin (2008)
187. McIlraith, S.A., Son, T.C., Zeng, H.: Semantic web services. IEEE Intell. Syst. **16**(2), 46–53 (2001)

188. Di Martino, B.: Semantic web services discovery based on structural ontology matching. Int. J. Web Grid Serv. **5**(1), 46–65 (2009)
189. Keller, U., Lara Hernandez, R., Fensel, D., Kifer, M., Polleres, A., Lausen, H., Zhao, C.: A logical framework for web service discovery. In: Proceedings of the ISWC 2004 Workshop: Semantic Web Services: Preparing to Meet the World of Business Applications, Hiroshima. CEUR Online Proceedings, CEUR-WS.org (2004). Accessed 23 Dec 2010
190. Kourtesis, D., Paraskakis, I.: Combining SAWSDL, OWL-DL and UDDI for semantically enhanced web service discovery. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) Proceedings of the Fifth European Semantic Web Conference (ESWC 2008), Tenerife. Lecture Notes in Computer Science, vol. 5021, pp. 614–628. Springer, Berlin (2008)
191. Klusch, M., Fries, B., Sycara, K.: Automated semantic web service discovery with OWLS-MX. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, pp. 915–922. ACM, New York (2006)
192. Klusch, M., Kaufer, F.: WSMO-MX: a hybrid semantic web service matchmaker. Web Intell. Agent Syst. **7**(1), 23–42 (2009)
193. Srivastava, B., Koehler, J.: Web service composition – current solutions and open problems. In: Workshop on Planning for Web Services at the 13th International Conference on Automated Planning & Scheduling (ICAPS 2003), Trento, pp. 28–35 (2003). http://www.isi.edu/integration/workshops/icaps2003-p4ws/. Accessed 23 Dec 2010
194. Hakimpour, F., Sell, D., Cabral, L., Domingue, J., Motta, E.: Semantic Web service composition in IRS-III: the structured approach. In: Proceedings of the Seventh IEEE International Conference on E-Commerce Technology (CEC 2005), Munich, pp. 484–487. IEEE Computer Society (2005)
195. Rao, J., Küngas, P., Matskin, M.: Composition of semantic web services using linear logic theorem proving. Inf. Syst. **31**(4), 340–360 (2006)
196. Lecue, F., Delteil, A., Leger, A.: Applying abduction in semantic web service composition. In: Proceedings of the IEEE International Conference on Web Services (ICWS-2007), Salt Lake City, pp. 94–101 (2007)
197. Lécué, F., Léger, A.: A formal model for semantic web service composition. In: Cruz, I.F., et al. (eds.) Proceedings of the Fifth International Semantic Web Conference (ISWC 2006), Athens, GA. Lecture Notes in Computer Science, vol. 4273, pp. 385–398. Springer, Heidelberg (2006)
198. Lécué, F., Léger, A., Delteil, A.: DL reasoning and AI planning for web service composition. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2008), Sydney, pp. 445–453 (2008)
199. Kim, J., Spraragen, M., Gil, Y.: An intelligent assistant for interactive workflow composition. In: Proceedings of the Ninth Internatinoal Conference on Intelligent User Interfaces (IUI 2004), Funchal, Madeira, pp. 125–131. ACM, New York (2004)
200. Zheng, G., Bouguettaya, A.: Discovering pathways of service oriented biological processes. In: Bailey, J., Maier, D., Schewe, K.-D., Thalheim, B., Wang, X.S. (eds.) Proceedings of the Ninth International Conference on Web Information Systems Engineering (WISE 2008), Auckland. Lecture Notes in Computer Science, vol. 5175, pp. 189–205. Springer, Berlin (2008)
201. Foster, I., Kesselman, C. (eds.): The GRID: Blueprint for a New Computing Infrastructure, 2nd edn. Morgan Kaufmann, San Francisco (2004)
202. De Roure, D., Jennings, N.R., Shadbolt, N.R.: The semantic grid: past, present, and future. Proc. IEEE **93**(3), 669–681 (2005). http://eprints.ecs.soton.ac.uk/9976/. Accessed 4 Aug 2010
203. Schollmeier, R.: A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In: Graham, R.L., Shahmehri, N. (eds.) Peer-to-Peer Computing, pp. 101–102. IEEE Computer Society, Washington, DC (2001)
204. Garcia-Molina, H., Crespo, A.: Semantic overlay networks for P2P systems. Technical report 2003-75, Stanford InfoLab. http://ilpubs.stanford.edu:8090/627/ (2003). Accessed 4 Aug 2010
205. Gu, T., Pung, H.K., Zhang, D.: Information retrieval in schema-based P2P systems using

one-dimensional semantic space. Comput. Netw. **51**(16), 4543–4560 (2007)

206. Gu, T., Pung, H.K., Zhang, D.: A semantic P2P framework for building context-aware applications in multiple smart spaces. In: Kuo, T.-W., Hsing-Mean Sha, E., Guo, M., Tianruo Yang, L., Shao, Z. (eds.) EUC. Lecture Notes in Computer Science, vol. 4808, pp. 553–564. Springer, Berlin (2007)
207. Aberer, K., Cudre-Mauroux, P., Hauswirth, M., van Pelt, T.: GridVine: building internet-scale semantic overlay networks. In: McIlraith, S.A., et al. (eds.) Proceedings of the Third International Semantic Web Conference (ISWC 2004), Hiroshima. Lecture Notes in Computer Science, vol. 3298, pp. 107–121. Springer, Berlin (2004)
208. Haase, P., Broekstra, J., Ehrig, M., Menken, M., Mika, P., Olko, M., Plechawski, M., Pyszlak, P., Schnizler, B., Siebes, R., Staab, S., Tempich, C.: Bibster – a semantics-based bibliographic peer-to-peer system. In: McIlraith, S.A., et al. (eds.) Proceedings of the Third International Semantic Web Conference (ISWC 2004), Hiroshima. Lecture Notes in Computer Science, vol. 3298, pp. 122–136. Springer, Berlin (2004)
209. Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M.: A break in the clouds: towards a cloud definition. SIGCOMM Comput. Commun. Rev. **39**(1), 50–55 (2009)
210. Youseff, L., Butrico, M., da Silva, D.: Towards a unified ontology of cloud computing. In: Grid Computing Environments Workshop (GCE08), held in conjunction with ACM/IEEE SuperComputing Conference (SC 2008), Austin (2008)
211. Mika, P., Tummarello, G.: Web semantics in the clouds. IEEE Intell. Syst. **23**(5), 82–87 (2008)
212. Mentzas, G., Friesen, A. (eds.): Semantic Enterprise Application Integration for Business Processes: Service-Oriented Frameworks. IGI Global, Hersley (2009)
213. Haniewicz, K., Kaczmarek, M., Zyskowski, D.: Semantic web services applications – a reality check. Wirtschaftsinformatik **50**(1), 39–46 (2008)
214. Klusch, M., Zhing, X.: Deployed semantic services for the common user of the web: a reality check. In: Proceedings of the International Conference on Semantic Computing (ICSC 2008), Santa Clara, pp. 347–353. IEEE Computer Society, Washington, DC (2008)
215. Roman, D., Klien, E.: SWING – a semantic framework for geospatial services. In: Scharl, A., Tochtermann, K. (eds.) The Geospatial Web: How Geo-Browsers, Social Software and the Web 2.0 are Shaping the Network Society. Springer, London (2007)
216. Zaharia, R., Vasiliu, L., Hoffman, J., Klien, E.: Semantic execution meets geospatial web services: a pilot application. Trans. GIS **12**(1), 59–73 (2009)
217. Dogac, A., Laleci, G.B., Kirbas, S., Kabak, Y., Sinir, S.S., Yildiz, A., Gurcan, Y.: Artemis: deploying semantically enriched web services in the healthcare domain. Inf. Syst. **31**(4), 321–339 (2006)
218. Schumacher, M., Helin, H., Schuldt, H. (eds.): CASCOM: Intelligent Service Coordination in the Semantic Web. Birkhäuser, Basel (2008)
219. della Valle, E., Zhao, G., Monteleone, G., Cerizza, D., Celino, I., Estublier, J., Vega, G., Kerrigan, M., Ramirez, J., Villazon-Terrazas, B., Guarrera, P.: SEEMP: a semantic interoperability infrastructure for e-Government services in the employment sector. In: Proceedings of the Fourth European Semantic Web Conference (ESWC 2007), Innsbruck. Lecture Notes in Computer Science, vol. 4519, pp. 220–234. Springer, Berlin (2007)
220. Gugliotta, A., Domingue, J., Cabral, L., Tanasescu, V., Galizia, S., Davies, R., Gutiérrez-Villarías, L., Rowlatt, M., Richardson, M., Stincic, S.: Deploying semantic web services-based applications in the e-Government domain. J. Data Semant. **10**, 96–132 (2008)
221. Milanovic, M., Gasevic, D., Kaviani, N.: Ontologies for software engineering. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, 2nd edn. Springer, Dordrecht (2009)
222. Tetlow, P., Pan, J.Z., Oberle, D., Wallace, E., Uschold, M., Kendall, E.: Ontology driven architectures and potential uses of the semantic web in systems and software engineering, W3C Working Draft, Working Group Note 2006/02/11, W3C. http://www.w3.org/2001/sw/BestPractices/SE/ODA/ (2006). Accessed 4 Aug 2010
223. Happel, H.-J., Seedorf, S.: Applications of ontologies in software engineering. In: Proceedings of the Workshop on Semantic Web Enabled

Software Engineering (SWESE 2006) at International Semantic Web Conference (ISWC 2006), Athens, GA (2006)

224. Kiko, K., Atkinson, C.: Integrating enterprise information representation languages. In: Proceedings of the International Workshop on Vocabularies, Ontologies and Rules for The Enterprise (VORTE 2005), Enschede, pp. 41–50 (2005)
225. Happel, H.-J., Korthaus, A., Seedorf, S., Tomczyk, P.: KOntoR: an ontology-enabled approach to software reuse. In: Zhang, K., Spanoudakis, G., Visaggio, G. (eds.) SEKE-2006, pp. 349–354 (2006)
226. Ankolekar, A., Sycara, K., Herbsleb, J., Kraut, R., Welty, C.: Supporting online problem-solving communities with the semantic web. In: Proceedings of the 15th International Conference on World Wide Web (WWW 2006), Edinburgh, pp. 575–584. ACM, New York (2006)
227. Dang, Q.V., Bac, C., Berger, O., Dao, X.S.: Improving community awareness in software forges by semantical aggregation of tools feeds. http://libresoft.es/oldsite/Activities/Research_activities/WoPDaSD2008_files/Paper4.pdf (2008). Accessed 4 Aug 2010
228. Doerr, M.: The CIDOC conceptual reference module: an ontological approach to semantic interoperability of metadata. AI Mag. **24**(3), 75–92 (2003)
229. Sinclair, P.A.S., Goodall, S., Lewis, P.H., Martinez, K., Addis, M.J.: Concept browsing for multimedia retrieval in the SCULPTEUR project. In: Proceedings of the Workshop on Multimedia and the Semantic Web at Second European Semantic Web Conference (ESWC 2005), Heraklion. Lecture Notes in Computer Science, vol. 3532. Springer, Berlin. http://eprints.ecs.soton.ac.uk/10913/ (2005). Accessed 4 Aug 2010
230. Vrochidis, S., Doulaverakis, C., Gounaris, A., Nidelkou, E., Makris, L., Kompatsiaris, I.: A hybrid ontology and visual-based retrieval model for cultural heritage multimedia collections. Int. J. Metadata Semant. Ontol. **3**(3), 167–182 (2008)
231. Hatala, M., Wakkary, R., Kalantari, L.: Rules and ontologies in support of real-time ubiquitous application. J. Web Semant. **3**(1), 5–22 (2005)
232. Abecker, A., Sheth, A., Mentzas, G., Stojanovic, L. (eds.) AAAI Spring Symposium Semantic Web Meets Egovernment. Technical Report SS-06-06. AAAI Press, Menlo Park (2006)
233. Charalabidis, Y., Metaxiotis, K.: Chapter XIII: ontology-based management of e-Government knowledge. In: Rahman, H. (ed.) Social and Political Implications of Data Mining: Knowledge Management in E-Government. IGI Global, Hershey (2008)
234. Crichton, C., Davies, J., Gibbons, J., Harris, S., Shukla, A.: Semantic frameworks for e-Government. In: Proceedings of the First International Conference on Theory and Practice of Electronic Governance (ICEGOV 2007), pp. 30–39. ACM (2007)
235. Stojanovic, L.: Ontology-based change management in an eGovernment application scenario. In: Studer, R., Grimm, S., Abecker, A. (eds.) Semantic Web Services: Concepts, Technologies, and Applications. Springer, Berlin (2008)
236. Ojo, A., Janowski, T., Estevez, E.: Semantic interoperability architecture for electronic government. In: Proceedings of the 10th Annual International Conference on Digital Government Research (DG.O 2009), Puebla, pp. 63–72. Digital Government Society of North America (2009)
237. Apostolou, D., Stojanovic, N., Anicic, D.: Responsive knowledge management for public administration: an event-driven approach. IEEE Intell. Syst. **24**(5), 20–30 (2009). Special Issue on Transforming E-government & E-participation
238. Sidoroff, T., Hyvönen, E.: Semantic e-government portals – a case study. In: Proceedings of the Workshop on Semantic Web Case Studies and Best Practices for eBusiness (SWCASE 2005) at Fourth International Semantic Web Conference (ISWC 2005), Galway (2005)
239. Trochidis, I., Tambouris, E., Tarabanis, K.A.: An ontology for modeling life-events. In: IEEE International Conference on Services Computing (SCC 2007), Salt Lake City, pp. 719–720. IEEE Computer Society, Washington, DC (2007)
240. Stevens, R., Goble, C.A., Bechhofer, S.: Ontology-based knowledge representation for bioinformatics. Brief. Bioinform. **1**(4), 398–414 (2000)

241. The Gene Ontology Consortium: Creating the gene ontology resource: design and implementation. Genome Res. **11**(8), 1425–1433 (2001)
242. Smith, B., Ashburner, M., Rosse, C., Bard, J., Bug, W., Ceusters, W., Goldberg, L.J., Eilbeck, K., Ireland, A., Mungall, C.J., Leontis, N., Rocca-Serra, P., Ruttenberg, A., Sansone, S.-A., Scheuermann, R.H., Shah, N., Whetzel, P.L., Lewis, S.: The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. Nat. Biotechnol. **25**(11), 1251–1255 (2007)
243. Kashyap, V., Cheung, K.-H., Doherty, D., Samwald, M., Marshall, M.S., Luciano, J., Stephens, S., Herman, I., Hookway, R.: Ontology-based data integration for biomedical research. In: Cardoso, J., et al. (eds.) The Semantic Web: Real-World Applications from Industry. Semantic Web and Beyond Computing for Human Experience, vol. 6, pp. 97–122. Springer, New York (2007)
244. Dietze, H., Schroeder, M.: GoWeb: a semantic search engine for the life Science Web. In: Burger, A., Paschke, A., Romano, P., Splendiani, A. (eds.) Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS 2008), Edinburgh. http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-435/ (2008). Accessed 4 Aug 2010
245. Bechhofer, S., Yesilada, Y., Stevens, R., Jupp, S., Horan, B.: Using ontologies and vocabularies for dynamic linking. IEEE Internet Comput. **12**(3), 32–39 (2008)
246. Rector, A.L., Qamar, R., Marley, T.: Binding ontologies and coding systems to electronic health records and messages. J. Appl. Ontol. **4**(1), 51–69 (2009)
247. Sheth, A.P., Agrawal, S., Lathem, J., Oldham, N., Wingate, H., Yadav, P., Gallagher, K.: Active semantic electronic medical records. In: Cardoso, J., et al. (eds.) The Semantic Web: Real-World Applications from Industry. Semantic Web And Beyond Computing for Human Experience, vol. 6, pp. 123–140. Springer, New York (2007)
248. Staab, S., Studer, R. (eds.): Handbook on Ontologies. International Handbooks on Information Systems, 2nd edn. Springer, Heidelberg (2009)
249. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC Press, Boca Raton (2009)
250. Allemang, D., Hendler, J.: Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL. Morgan Kaufmann, San Francisco (2008)
251. d'Aquin, M., Castro, A.G., Lange, C., Viljanen, K. (eds.) Proceedings of the First International Workshop on Ontology Repository and Editors for the Semantic Work (ORES 2010) at Seventh Extended Semantic Web Conference (ESWC 2010), Heraklion. Online Proceedings: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-596/(2010). Accessed 10 Aug 2010
252. Buitelaar, P., Eigner, T.: Evaluating ontology search. In: Garcia-Castro, R., Vrandecic, D., Gómez-Pérez, A., Sure, Y., Huang, Z. (eds.) Proceedings of the Fifth International Workshop on Evaluation of Ontologies and Ontology-based Tools (EON 2007), co-located with Sixth International Semantic Web Conference (ISWC 2007), Busan. CEUR Workshop Proceedings, vol. 329, pp. 11–20. CEUR-WS.org (2007). Accessed 1 Jan 2011
253. Rubin, D.L., Moreira, D.A., Kanjamala, P.P., Musen, M.A.: BioPortal: a web portal to biomedical ontologies. In: Sleeman, D., Musen, M. (eds.) AAAI 2008 Spring Symposium on Symbiotic Relationships between Semantic Web and Knowledge Engineering. Stanford University, Palo Alto (2008)
254. Goble, C.A., Corcho, O., Alper, P., de Roure, D.: e-Science and the Semantic Web: a symbiotic relationship. In: Proceedings of the 17th International Conference on Algorithmic Learning Theory (ALT 2006), Barcelona (2006)
255. Codina, J., Pianta, E., Vrochidis, S., Papadopoulos, S.: Integration of semantic, metadata and image search engines with a text search engine for patent retrieval. In: Bloehdorn, S., Grobelnik, M., Mika, P., Tran, D.T. (eds.) Workshop on Semantic Search at Fifth European Semantic Web Conference 2008 (ESWC 2008), Tenerife. Lecture Notes in Computer Science, vol. 5021, pp. 14–28. Springer, Berlin (2008)
256. van Elst, L., Abecker, A.: Ontologies for information management: balancing formality,

stability, and sharing scope. Expert Syst. Appl. **23**(4), 357–366 (2002)

257. Hepp, M.: Possible ontologies: how reality constrains the development of relevant ontologies. IEEE Internet Comput. **11**(1), 90–96 (2007)
258. Zacharias, V., Braun, S., Schmidt, A.: Social semantic bookmarking with SOBOLEO. In: Murugesan, S. (ed.) Handbook of Research on Web 2.0, 3.0 and X.0: Technologies, Business, and Social Applications. IGI Global, Hershey (2009)
259. Wang, H., Liu, Q., Penin, T., Fu, L., Zhang, L., Thanh Tran, D., Yu, Y., Pan, Y.: Semplore: a scalable IR approach to search the web of data. J. Web Semant. **7**(3), 177–188 (2009)
260. Thanh Tran, D., Haase, P., Studer, R.: Semantic search – using graph-structured semantic models for supporting the search process. In: Proceedings of the 17th International Conference on Conceptual Structures (ICCS 2009), Moscow (2009)
261. Chen, H., Finin, T., Joshi, A.: An ontology for context-aware pervasive computing environments. Knowl. Eng. Rev. **18**(3), 197–207 (2003)
262. Gu, T., Wang, X.H., Pung, H.K., Zhang, D.Q.: An ontology-based context model in intelligent environments. In: Proceedings of the 2004 Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2004), San Diego, pp. 270–275 (2004)
263. Corcho, O., Hauswirth, M., Koubarakis, M. (eds.) Proceedings of the International Workshop on the Semanticm Sensor Web (SemSensWeb 2009), Collocated with ESWC 2009, Crete. CEUR Online Proceedings. http://www.ceur-ws.org/Vol-468 (2009). Accessed 4 Aug 2010
264. Sheth, A., Henson, C., Sahoo, S.S.: Semantic sensor web. IEEE Internet Comput. **12**(4), 78–83 (2008)
265. Cardoso, J., Hepp, M., Lytras, M.D. (eds.): The Semantic Web: Real-World Applications from Industry. Semantic Web and Beyond Computing for Human Experience, vol. 6. Springer, New York (2007)
266. Cruz, I.F., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L. (eds.): Proceedings of the Fifth International Semantic Web Conference (ISWC 2006), Athens, GA. Lecture Notes in Computer Science, vol. 4273. Springer, Heidelberg (2006)
267. McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.): Proceedings of the Third International Semantic Web Conference (ISWC 2004), Hiroshima. Lecture Notes in Computer Science, vol. 3298. Springer, Heidelburg (2004)