2

Ontology-based information sharing

Summary. In the last chapter we introduced the general problem of information sharing in the presence of heterogeneous data. In this chapter, we introduce ontologies as a means of dealing with semantic heterogeneity. We discuss the nature and applications of ontologies and review existing approaches that use ontologies for dealing with heterogeneous data. We also identify the state of the art in ontology-based information integration and identify open problems that will be addressed in the remainder of the book.

As we have seen in the last chapter, intelligent information sharing needs explicit representations of information semantics. We reviewed different approaches for capturing semantics that have been developed in different scientific communities. In this section we discuss ontologies as a general mechanism for representing information semantics that can be implemented using the approaches mentioned in Chap. 1. We start with a general introduction to the notion of ontologies and argue for their benefits for information integration and retrieval making them suitable as a tool for supporting information sharing. We also review the use of ontologies in the information-integration literature identifying ontology-based architectures for information sharing. Based on the review of integration architectures we present a general framework for supporting information sharing on the semantic web that summarizes the work reported in the remainder of the book. We relate the framework to existing work and give pointers to the different chapters of the book. Finally, we describe the representational infrastructure that is the core feature of the framework.

2.1 Ontologies

In this section we argue for ontologies as a technology for approaching the problem of explicating semantic knowledge about information. We first give a

general overview of the nature and purpose of ontologies that already reveals a great potential with respect to our task. Afterwards we sketch the idea of how ontologies could be used in order to support the semantic translation process. The idea presented will be elaborated in the remainder of the book.

The term "ontology" has been used in many ways and across different communities [Guarino and Giaretta, 1995]. If we want to motivate the use of ontologies for geographic information processing we have to make clear what we have in mind when we refer to ontologies. Thereby we mainly follow the description given in [Uschold and Gruninger, 1996]. In the following sections we will introduce ontologies as an explication of some shared vocabulary or conceptualization of a specific subject matter. We will briefly describe the way an ontology explicates concepts and their properties and argue for the benefit of this explication in different typical application scenarios.

2.1.1 Shared vocabularies and conceptualizations

In general, each person has her individual views on the World and the things she has to deal with every day. However, there is a common basis of understanding in terms of the language we use to communicate with each other. Terms from natural language can therefore be assumed to be a shared vocabulary relying on a (mostly) common understanding of certain concepts with only little variety. This common understanding relies on the idea of how the World is organized. We often call this idea a "conceptualization" of the World. Such conceptualizations provide a terminology that can be used for communication.

The example of natural language already shows that a conceptualization is never universally valid, but rather for a limited number of persons committing to that conceptualization. This fact is reflected in the existence of different languages which differ more or less. For example, Dutch and German share many terms; however, Dutch contains far more terms for describing bodies of water, due to the great importance of water in the life of people. Things get even worse when we are not concerned with everyday language but with terminologies developed for special areas. In these cases we often find situations where the same term refers to different phenomena. The use of the term "ontology" in philosophy and its use in computer science may serve as an example. The consequence is a separation into different groups that share a terminology and its conceptualization. These groups are also called information communities [Kottmann, 1999] or ontology groups [Fensel et al., 1997]. An example of such a community is the $(KA)^2$ initiative [Benjamins and Fensel, 1998].

The main problem with the use of a shared terminology according to a specific conceptualization of the World is that much information remains implicit.

When a mathematician talks about the binomial $\binom{n}{k}$ he has much more in mind than just the formula itself. He will also think about its interpretation (the number of subsets of a certain size) and its potential uses (e.g. estimating the chance of winning in a lottery). Ontologies have set out to overcome the problem of implicit and hidden knowledge by making the conceptualization of a domain (e.g. mathematics) explicit. This corresponds to one of the definitions of the term ontology most popular in computer science [Gruber, 1993]:

"An ontology is an explicit specification of a conceptualization."

An ontology is used to make assumptions about the meaning of a term available. It can also be seen as an explication of the context a term is normally used in. Lenat [Lenat, 1998] for example describes context in terms of 12 independent dimensions that have to be known in order to understand a piece of knowledge completely and shows how these dimensions can be explicated using the Cyc ontology.

2.1.2 Specification of context knowledge

There are many different ways in which an ontology may explicate a conceptualization and the corresponding context knowledge. The possibilities range from a purely informal natural-language description of a term corresponding to a glossary up to strictly formal approaches with the expressive power of full first-order predicate logic or even beyond (e.g. ONTOLINGUA [Gruber, 1991]). Jasper and Uschold distinguish two ways in which the mechanisms for the specification of context knowledge by an ontology can be compared [Jasper and Uschold, 1999]:

Level of formality

The specification of a conceptualization and its implicit context knowledge can be done at different levels of formality. As already mentioned above, a glossary of terms can also be seen as an ontology despite its purely informal character. A first step to gain more formality is to prescribe a structure to be used for the description. A good example for this approach is the new standard Web annotation language XML [Yergeau et al., 2004]. XML offers the possibility to define terms and organize them in a simple hierarchy according to the expected structure of the Web document to be described in XML. However, the rather informal character of XML encourages its misuse. While the hierarchy of an XML specification was originally designed to describe layout it can also be exploited to represent subtype hierarchies [van Harmelen and Fensel, 1999], which may lead to confusion. This problem can be solved by assigning formal semantics to the structures used for the description of the ontology. An example is the conceptual modelling language CML [Schreiber et al., 1994]. CML offers primitives to describe a domain that can be given a formal semantics in terms of first order logic [Aben, 1993]. However, a formalization is

only available for the structural part of a specification. Assertions about terms and the description of dynamic knowledge are not formalized, offering total freedom for the description. On the other extreme there are also specification languages which are completely formal. A prominent example is ONTOLINGUA (see above), one of the first Ontology languages which is based on the knowledge interchange format KIF [Genesereth and Fikes, 1992] which was designed to enable different knowledge-based systems to exchange knowledge.

Extent of Explication

The other comparison criterion is the extent of explication that is reached by the ontology. Jasper and Uschold [Jasper and Uschold, 1999] refer to "lightweight" vs. "heavyweight" ontologies to described differences in the extend of explication. This criterion is strongly connected with the expressive power of the specification language used. We can generalize this by saying that the least expressive specification of an ontology consists of an organization of terms in a network using two-placed relations. This idea goes back to the use of semantic networks. Many extensions of the basic idea have been proposed. One of the most influential was the use of roles that could be filled out by entities showing a certain type [Brachman, 1977]. This kind of value restriction can still be found in recent approaches. RDF Schema descriptions [Brickley and Guha, 2004] (see Chap. 3 which is the new standard for the semantic descriptions of Web pages, is an example. An RDF Schema contains class definitions with associated properties that can be restricted by so-called constraint-properties. However, default values and value-range descriptions are not expressive enough to cover all possible conceptualizations. A greater expressive power can be provided by allowing classes to be specified by logical formulas. These formulas can be restricted to a decidable subset of first order logic. This is the approach of so-called description logics [Donini et al., 1996]. This trade-off between expressiveness and decidability is also reflected in the development of the Web Ontology Language OWL which is described in more details in Chap. 3 where the language subset that corresponds to description logics is explicitly distinguished. Nevertheless, there are also approaches allowing for more expressive descriptions. In ONTOLINGUA, for example, classes can be defined by arbitrary KIF expressions. Beyond the expressiveness of full first-order predicate logic there are also special purpose languages that have an extended expressiveness to cover specific needs of their application area. The latest example is OWL, where the complete language (OWL full) is undecidable[1] as it combines description logics with meta-level features.

[1] undecidability still has to be proven formally, but there are no doubts about this fact

2.1.3 Beneficial applications

Ontologies are useful for many different applications that can be classified into several areas [Jasper and Uschold, 1999]. Each of these areas has different requirements on the level of formality and the extent of explication provided by the ontology. The common idea of all of these applications is to use ontologies in order to reach a common understanding of a particular domain. In contrast to syntactic standards, the understanding is not restricted to a common representation or a common structure. The use of ontologies also helps to reach a common understanding of the *meaning* of terms. Therefore, ontologies are a promising candidate in order to support semantic interoperability. We will shortly review some common application areas, namely the support of communication processes, the specification of systems and information entities and the interoperability of computer systems.

Communication

Information communities are useful, because they ease communication and cooperation among their members by the use of a shared terminology with a well-defined meaning. On the other hand, the formation of information communities makes communication between members from different information communities very difficult, because they do not agree on a common conceptualization. They may use the shared vocabulary of natural language. However, most of the vocabulary used in their information communities is highly specialized and not shared with other communities. This situation demands an explication and explanation of the terminology used. Informal ontologies with a large extent of explication are a good choice to overcome these problems. While definitions have always played an important role in scientific literature, conceptual models of certain domains are rather new. However, nowadays systems analysis and related fields like software engineering rely on conceptual modelling to communicate structure and details of a problem domain as well as the proposed solution between domain experts and engineers. Prominent examples of ontologies used for communication are entity-relationship diagrams [Chen, 1976] and object-oriented modelling languages like UML [Rumbaugh et al., 1998].

Systems engineering

Entity-relationship diagrams as well as UML are not only used for communication, they also serve as building plans for data and systems guiding the process of building (engineering) the system. The use of ontologies for the description of information and systems has many benefits. The ontology can be used to identify requirements as well as inconsistencies in a chosen design. It can help to acquire or search for available information. Once a systems component has been implemented its specification can be used for maintenance

and extension purposes. Another very challenging application of ontology-based specification is the re-use of existing software. In this case the specifying ontology serves as a basis to decide if an existing component matches the requirements of a given task [Motta, 1999]. Depending on the purpose of the specification, ontologies of different formal strength and expressiveness are to be used. While the process of communicating design decisions and the acquisition of additional information normally benefit from rather informal and expressive ontology representations (often graphical), the directed search for information needs a rather strict specification with a limited vocabulary to limit the computational effort. At the moment, the support of semi- automatic software re-use seems to be one of the most challenging applications of ontologies, because it requires expressive ontologies with a high level of formal strength (see for example [van Heijst et al., 1997]).

Interoperability

The above considerations might provoke the impression that the benefits of ontologies are limited to systems analysis and design. However, an important application area of ontologies is the integration of existing systems. The ability to exchange information at run time, also known as interoperability, is an important topic. The attempt to provide interoperability suffers from problems similar to those associated with the communication amongst different information communities. The important difference is that the actors are not persons able to perform abstraction and common sense reasoning about the meaning of terms, but machines. In order to enable machines to understand each other we also have to explicate the context of each system, but on a much higher level of formality in order to make it machine understandable (the KIF language was originally defined for the purpose of exchanging knowledge models between different knowledge-based systems). Ontologies are often used as interlinguas for providing interoperability [Uschold and Gruninger, 1996]: they serve as a common format for data interchange. Each system that wants to interoperate with other systems has to transfer its information into this common framework.

Information Retrieval

Common information-retrieval techniques either rely on a specific encoding of available information (e.g. fixed classification codes) or simple full-text analysis. Both approaches suffer from severe shortcomings. First of all, both completely rely on the input vocabulary of the user, which might not be completely consistent with the vocabulary of the information. Second, a specific encoding significantly reduces the recall of a query, because related information with a slightly different encoding is not matched. Full-text analysis on the other hand reduces precision, because the meaning of the words might be ambiguous.

Using an ontology in order to explicate the vocabulary can help overcome some of these problems. When used for the description of available information as well as for query formulation, an ontology serves as a common basis for matching queries against potential results on a semantic level. The use of rather informal ontologies like WordNet [Fellbaum, 1998] increases the recall of a query by including synonyms in the search process. The use of more formal representations like conceptual graphs [Sowa, 1999] further enhances the retrieval process, because a formal representation can be used to increase recall by reasoning about inheritance relationships and precision by matching structures. To summarize, information retrieval benefits from the use of ontologies. Ontologies help to decouple description and query vocabularies and increase precision as well as recall [Guarino et al., 1999].

2.2 Ontologies in information integration

We analyzed about 25 approaches to intelligent information integration including SIMS [Arens et al., 1993], TSIMMIS [Garcia-Molina et al., 1995], OBSERVER [Mena et al., 2000a] , CARNOT [Collet et al., 1991], Infosleuth [Nodine et al., 1999], KRAFT [Preece et al., 1999], PICSEL [Levy et al., 1996], DWQ [Calvanese et al., 1998b], Ontobroker [Fensel et al., 1998] , SHOE [Heflin et al., 1999] and others with respect to the role and use of ontologies. While all of the systems used ontologies to describe the meaning of information, the role and use of these descriptions differ between the approaches. In the following we discuss the different roles ontologies can play in information integration.

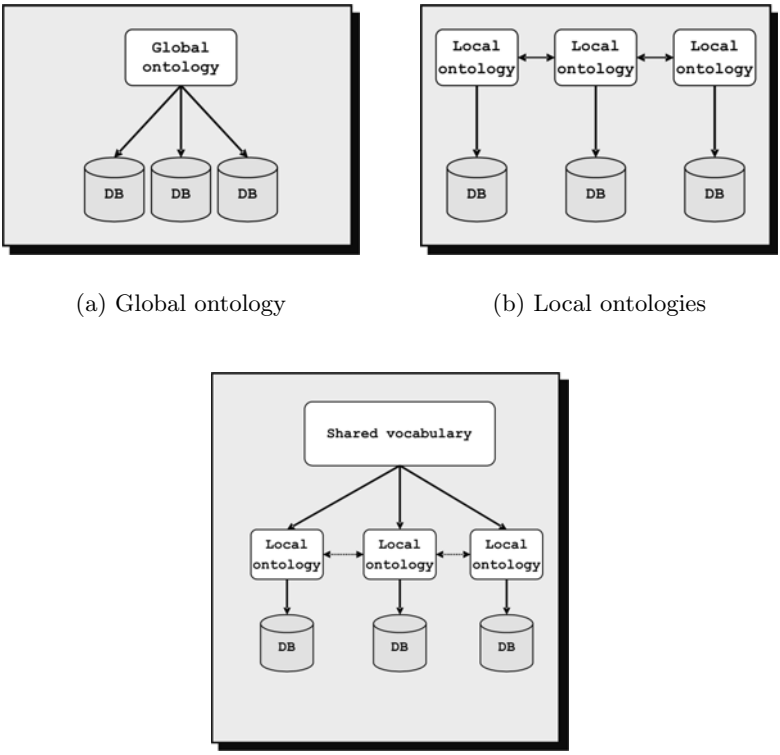
2.2.1 Content explication

In nearly all ontology-based integration approaches ontologies are used for the explicit description of the information-source semantics. But there are different ways of how to employ the ontologies. In general, three different directions can be identified: *single-ontology approaches*, *multiple-ontology approaches* and *hybrid approaches*. Fig. 2.1 gives an overview of the three main architectures.

The integration based on a single ontology seems to be the simplest approach because it can be simulated by the other approaches. Some approaches provide a general framework where all three architectures can be implemented (e.g. DWQ [Calvanese et al., 1998b]). The following paragraphs give a brief overview of the three main ontology architectures.

Single-ontology approaches

Single-ontology approaches use one global ontology providing a shared vocabulary for the specification of the semantics (see Fig. 2.1a). All information



(a) Global ontology

(b) Local ontologies

(c) Hybrid approach

Fig. 2.1. The three possible ways for using ontologies for content explication

sources are related to the one global ontology. A prominent approach of this kind of ontology integration is SIMS [Arens et al., 1993]. SIMS model of the application domain includes a hierarchical terminological knowledge base with nodes representing objects, actions and states. An independent model of each information source must be described for this system by relating the objects of each source to the global domain model. The relationships clarify the semantics of the source objects and help to find semantically corresponding objects. The global ontology can also be a combination of several specialized ontologies. A reason for the combination of several ontologies can be the modularization of a potentially large monolithic ontology. The combination is supported by ontology-representation formalisms, i.e. importing other ontology modules (cf. ONTOLINGUA [Gruber, 1991]).

Single-ontology approaches can be applied to integration problems where all information sources to be integrated provide nearly the same view of a domain. But if one information source has a different view of a domain, e.g. by providing another level of granularity, finding the minimal ontology commitment [Gruber, 1995] becomes a difficult task. For example, if two information sources provide product specifications but refer to absolute heterogeneous product catalogues which categorize the products, the development of a global ontology which combines the different product catalogues becomes very difficult. Information sources with reference to similar product catalogues are much easier to integrate. Also, single-ontology approaches are susceptible to changes in the information sources, which can affect the conceptualization of the domain represented in the ontology. Depending on the nature of the changes in one information source it can imply changes in the global ontology and in the mappings to the other information sources. These disadvantages led to the development of multiple-ontology approaches.

Multiple ontologies

In multiple-ontology approaches, each information source is described by its own ontology (Fig. 2.1b). For example, in OBSERVER [Mena et al., 2000a], the semantics of an information source is described by a separate ontology. In principle, the "source ontology" can be a combination of several other ontologies but it cannot be assumed that the different "source ontologies" share the same vocabulary.

At a first glance, the advantage of multiple-ontology approaches seems to be that no common and minimal ontology commitment [Gruber, 1995] about one global ontology is needed. Each source ontology could be developed without reference to the other sources or their ontologies, no common ontology with the agreement of all sources is needed. This ontology architecture can simplify the change, i.e. modifications in one information source or the adding and removing of sources. But in reality the lack of a common vocabulary makes it extremely difficult to compare different source ontologies. To overcome this problem, an additional representation formalism defining the mapping is provided. The mapping identifies semantically corresponding terms of different source ontologies, e.g. which terms are semantically equal or similar. But the mapping also has to consider different views of a domain, e.g. different aggregation and granularity of the ontology concepts. We believe that in practice the mapping is very difficult to define, because of the many semantic heterogeneity problems which may occur.

Hybrid approaches

To overcome the drawbacks of the single- or multiple-ontology approaches, hybrid approaches were developed (Fig. 2.1c). Similar to multiple-ontology approaches the semantics of each source is described by its own ontology.

But in order to make the source ontologies comparable to each other they are built upon one global shared vocabulary [Wache et al., 1999, Goh, 1997]. The shared vocabulary contains basic terms (the primitives) of a domain. In order to build complex terms of a source ontology the primitives are combined by some operators. Because each term of a source ontology is based on the primitives, the terms become easier comparable than in multiple-ontology approaches. Sometimes the shared vocabulary is also an ontology [Stuckenschmidt and Wache, 2000].

In hybrid approaches the interesting point is how the local ontologies are described, i.e. how the terms of the source ontology are described by the primitives of the shared vocabulary.

- In COIN [Goh, 1997], the local description of an information, the so-called context, is simply an attribute value vector. The terms for the context stems from the common shared vocabulary and the data itself.
- In MECOTA [Wache, 1999], each source information is annotated by a label which indicates the semantics of the information. The label combines the primitive terms from the shared vocabulary. The combination operators are similar to the operators known from the description logics, but are extended for the special requirements resulting from integration of sources, e.g. by an operator which indicates that an information aggregates several different information items (e.g. a street name together with a number).
- In BUSTER [Visser et al., 2002], the shared vocabulary is a (general) ontology, which covers all possible refinements. The general ontology may define the attribute value ranges of its concepts. A source ontology is one (partial) refinement of the general ontology, e.g. it restricts the value range of some attributes. Since the source ontologies only use the vocabulary of the general ontology, they remain comparable.

The advantage of a hybrid approach is that new sources can easily be added without the need of modification in the mappings or in the shared vocabulary. It also supports the acquisition and evolution of ontologies. The use of a shared vocabulary makes the source ontologies comparable and avoids the disadvantages of multiple-ontology approaches. The drawback of hybrid approaches , however, is that existing ontologies cannot be re-used easily, but have to be re-developed from scratch, because all source ontologies have to refer to the shared vocabulary. Table 2.1 summarizes the benefits and drawbacks of the different ontology approaches.

2.2.2 Additional roles of ontologies

Some approaches use ontologies not only for content explication, but also either as a global query model or for the verification of the (user-defined or

Table 2.1. Comparison of ontology-based integration approaches

	Single-ontology approaches	Multiple-ontology approaches	Hybrid approaches
Implementation effort	Straightforward	Costly	Reasonable
Semantic heterogeneity	Similar views of a domain	Supports heterogeneous views	Supports heterogeneous views
Adding/removing sources	Need for some adaption in the global ontology	Providing a new source ontology; relating to other ontologies	Providing a new source ontology
Comparing multiple ontologies	—	Difficult because of the lack of a common vocabulary	Simple because ontologies use a common vocabulary

system-generated) integration description. In the following, these additional roles of ontologies are considered in more detail.

Query model

Integrated information sources normally provide an integrated global view. Some integration approaches use the ontology as the global query schema. For example, in SIMS [Arens et al., 1996] the user formulates a query in terms of the ontology. Then SIMS reformulates the global query into subqueries for each appropriate source, collects and combines the query results and returns the results. The use of ontologies as query models is independent of the use of a global ontology. In OBSERVER, for example, the user can pose queries using terms from the ontology of the local source.

Using an ontology as a query model has the advantage that the structure of the query model should be more intuitive for the user because it corresponds more to the user's appreciation of the domain. But from a database point of view this ontology only acts as a global query schema. If a user formulates a query, he has to know the structure and the contents of the ontology; he cannot formulate the query according to a schema he would prefer personally. Therefore, it is questionable whether the global ontology is an appropriate query model.

Verification

During the integration process several mappings must be specified from a global schema to the local source schema. The correctness of such mappings can be considered ably improved if these can be verified automatically. A subquery is correct with respect to a global query if the local subquery provides a part of the queried answers, i.e. the subqueries must be contained in the global query (query containment) [Goasdoue et al., 2000, Calvanese et al., 1998a]. Since an ontology contains a (complete) specification of the conceptualization, the mappings can be validated with respect to the ontologies. Query containment means that the ontology concepts corresponding to the local sub-queries are contained in the ontology concepts related to the global query.

In DWQ [Calvanese et al., 1998b], each source is assumed to be a collection of relational tables. Each table is described in terms of its ontology with the help of conjunctive queries. A global query and the decomposed subqueries can be unfolded to their ontology concepts. The subqueries are correct, i.e. are contained in the global query, if their ontology concepts are subsumed by the global ontology concepts. The PICSEL project [Goasdoue et al., 2000] can also verify the mapping, but in contrast to DWQ it can also generate mapping hypotheses automatically which are validated with respect to a global ontology.

The quality of the verification task strongly depends on the completeness of an ontology. If the ontology is incomplete, the verification result can erroneously imagine a correct query subsumption. Since in general the completeness cannot be measured, it is impossible to make any statements about the quality of the verification.

2.3 A framework for information sharing

In this book, we describe different components of a framework for information sharing on the Semantic Web. The design of the framework is motivated by the potential roles of ontologies in information integration. In particular, we use ontologies to represent the intended interpretation of contents different information sources. We adopt the hybrid approach because it provides a good trade-off with respect to development costs and maintainability. We assume that a shared vocabulary provides the foundation for query formulation, for translations between the ontologies describing different information sources and for the verification of metadata as well as mappings between sources. Taking the hybrid approach as a starting point, our framework contains three main components whose relations are sketched in Fig. 2.2. In the following, we briefly describe the different components, their relations and related them to parts of the book.

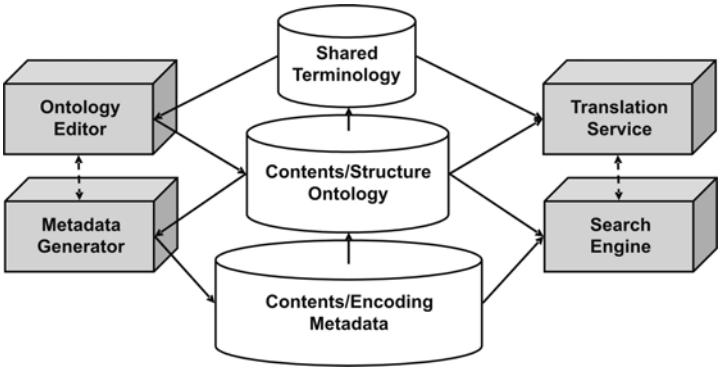


Fig. 2.2. A framework for ontology-based information sharing

Representational Infrastructure

The representational infrastructure we use to facilitate information sharing consists of three layers. On the lowest level, metadata describes the content of information sources. We assume that this metadata is represented using RDF as a common language. On the highest level, a common vocabulary defines terms that are assumed to have the same meaning across all information sources. This shared vocabulary mostly consists of concept hierarchies and relations between concepts in these hierarchies. We use RDF Schema to represent this information. Unlike most current work in the semantic web area, we do not directly layer RDF Schema models on top of the RDF metadata. Instead we insert an additional layer consisting of ontologies that describe the conceptualizations and structures specific to a certain source of information. The definitions in these ontologies are built using terms from the shared vocabulary. We encode these source-specific ontologies using the web ontology language OWL. The expressive power of OWL allows us to accurately define the intended meaning of the modelling elements and the data values used in the different sources. After providing a more detailed description of this layering in the remainder of this chapter, we introduce RDF Schema and OWL in Chap. 3. We also present an extension to OWL that allows the definition of direct mappings between source ontologies in Chap. 10 and describe source ontologies and metadata models for special data sources, more specifically for statistical (Chap. 7) and for spatially-related information (Chap. 8).

Development and Maintenance

In order to be able to share information from different sources it is not enough to described the representational infrastructure needed, we also have to address the problem of creating the infrastructure. In particular, this includes the selection of a shared vocabulary, the definition of source ontologies as well as the generation of metadata description for the concrete information in a

source. We envision this process to start with the analysis of the information sources to be shared and the conceptual choices made therein. Based on the these conceptualizations and the kind of information to be shared, candidates for a shared vocabulary are selected and refined in an iterative process which is supported by standard editing tools for OWL and RDF Schema. We further assume that metadata for the different sources is created independently of each other using the source ontology that has been build before. We developed tools for creating metadata by discovering patterns in the structure of information sources and link them to concepts in the source ontology. Finally, we describe a framework for managing the evolution of ontologies that are linked by mappings in order to react to changes in the information. We discuss the creation of source ontologies and shared vocabulary along with the detailed description of their connection in Chap. 4. The automatic creation of metadata based on the result of this process is described in Chap. 5. Evolution management for interrelated ontologies is the topic of Chap. 11.

Retrieval and Integration

The ultimate goal of our framework is to enable people to share information across different sources in a meaningful way. The representational infrastructure for describing information semantics and the methods for building and maintaining these representations are a necessary pre-condition for approach this goal. Based on this infrastructure our framework provides two principled mechanisms for supporting information sharing: Methods for content-based retrieval of information from remote sources and methods for translating between the conceptualizations of different sources. The translation services, we describe in this book are mainly concerned with domain conflicts by detecting and resolving conflicts between the the definition of object classes. Our methods exploit the existence of a shared vocabulary and uses existing reasoning systems for OWL and RDF schema to automatically compute subsumption relations between from different ontologies. The translation and the retrieval methods are tightly integrated as translation is needed during the retrieval process in order to find relevant information and for actually translating retrieved data items into the terminology used by the user. The retrieval and integration methods are described in Chap. 6. Large parts of these methods have been implemented in the BUSTER system which follows the schema in Fig. 2.2. On the other hand, similar methods are found in other existing systems. Three of these systems that all more or less implement parts of the framework are described in Chap. 9.

2.4 A translation approach to ontology alignment

The core idea of the information framework sketched above is the use of a shared vocabulary as a basis for comparing the conceptualizations of different

information sources. The existence of such a shared vocabulary makes it possible to translate between different information on a semantic level. On the Semantic Web, it will frequently happen that information sources are added or removed. Further, the number of information sources will be considerably high. Based on these observations, we conclude that an on-demand translation of information semantics is most adequate for our purposes. Therefore, we will use the idea of integration by translation as a guideline for the remainder of the book.

2.4.1 The translation process

The proposed translation process is sketched below describing actors, supporting tools and knowledge items (i.e. ontologies) involved. Notice that although the approach described above translates only between two sources at a time, it is not limited to bilateral integration.

Authoring of shared terminology

Our approach relies on the use of a shared terminology in terms of properties used to define different concepts. This shared terminology has to be general enough to be used across all information sources to be integrated but specific enough to make meaningful definitions possible. Therefore the shared terminology will normally be built by an independent domain expert who is familiar with typical tasks and problems in a domain, but who is not concerned with a specific information source. As building a domain ontology is a challenging task, sufficient tool support has to be provided to build that ontology. A growing number of ontology editors exist [Duineveld et al., 1999]. The choice of a tool has to be based on the special needs of the domain to be modelled and the knowledge of the expert.

Annotation of information sources

Once a common vocabulary exists, it can be used to annotate different information sources. In this case annotation means that the inherent concept hierarchy of an information source is extracted and each concept is described by necessary and sufficient conditions using the terminology built in step one. The result of this annotation process is an ontology of the information source to be integrated. The annotation will normally be done by the owner of an information source who wants to provide better access to his or her information. In order to enable the information owner to annotate his information he has to know about the right vocabulary to use. It will also be beneficial to provide tool support also for this step. We need an annotation tool with different repositories of vocabularies according to different domains of interest.

Semantic translation of information entities

The only purpose of the steps described above was to lay a base for the actual translation step. The existence of ontologies for all information sources to be integrated enables the translator to work on these ontologies instead of treating real data. This way of using ontologies as surrogates for information sources has already been investigated in the context of information retrieval [Visser and Stuckenschmidt, 1999]. In that paper we showed that the search for interesting information can be enhanced by ontologies. Concerning semantic translation the use of ontologies as surrogates for information sources enables us to restrict the translation to the transformation of type information attached to an information entity by manipulating concept terms indicating the type of the entity.

The new concept term describing the type of an information entity in the target information source is determined automatically by an inference engine that uses ontologies of source and target structures as classification knowledge. This is possible, because both ontologies are based on the same basic vocabulary that has been built in the first step of the integration approach.

2.4.2 Required infrastructure

In order to enable a terminological reasoning system to actually relate concepts, we have to make assumptions about the knowledge represented. These assumptions directly refer to the two solutions to the explication dilemma mentioned above, because reasoning across ontologies requires a shared basic vocabulary (reduction to syntax) and the description of concepts in both ontologies in terms of logical expressions over these shared terms (reduction to logic).

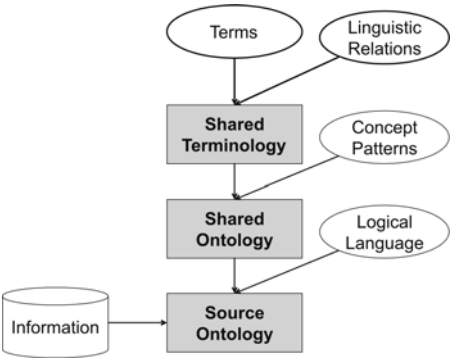


Fig. 2.3. Conceptual Architecture of the Translation Knowledge

We distinguish between shared terminology and shared ontology. The shared terminology consists of terms assumed to have a unique meaning across different classifications. These terms are structured by relations borrowed from linguistics, i.e. synonym (equivalent term), hypernym (broader term) and hyponym (more specialized term) relations. Formally, we define a shared terminology as a set of words and a partial function over pairs of words:

Definition 2.1 (shared terminology). *A shared terminology is a tuple* $\langle W, l\rangle$*, where* W *is a set of words and* $l : W \times W \rightarrow \{syno, hyper, hypo\}$ *is a partial function from the set of all pairs of terms into a set of identifiers specifying whether the first term is a synonym, a hypernym or a hyponym of the second.*

This shared terminology is linked to the specific integration problem using *structural patterns*. A structural pattern is a general specification of relations between objects denoted by the words in the shared terminology. In order to be able to apply these relations to information objects, the shared terminology is encoded in a simple logical structure consisting of a set of terms corresponding to words from the shared terminology relations between these terms and a set of axioms. The axioms define the synonym, hypernym and hyponym relations between terms in terms of the subsumption relation.

Definition 2.2 (shared ontology). *A shared ontology is a tuple* $\langle ST, T, R, A\rangle$*, where* $ST = \langle W_L, l_L\rangle$ *is a shared terminology,* T *is a basic set of terms,* R *is a set of relations* $R \subseteq T \times T$ *and* A *is a set of axioms of the form* $T_i \sqsubseteq T_j$ *if the following conditions hold:*

- $T \subseteq W_L$*,*
- *for each pair of words* (W_i, W_j)*,*
  - *if* $l((W_i, W_j)) = hyper$ *then* $W_j \sqsubseteq W_i$ *is in* A*,*
  - *if* $l((W_i, W_j)) = hypo$ *then* $W_i \sqsubseteq W_j$ *is in* A*,*
  - *if* $l((W_i, W_j)) = syno$ *then* $W_i \sqsubseteq W_j$ *and* $W_j \sqsubseteq W_i$ *are in* A*.*

From the point of expressiveness, this shared ontology is very similar to a model in RDF Schema, because it defines a hierarchy of terms (classes in RDF Schema) as well as a set of relations (properties) with corresponding range and domain restrictions. This correspondence enables us to use RDF Schema in order to encode shared ontologies as a basis for defining information semantics.

Shared ontologies provide us with a vocabulary we can use in order to specify the semantics of information in different sources. This semantics, however, has to be defined individually for different information sources. In order to capture the semantics of types or assessments used in an information source, we need a richer language, because their meaning almost never directly corresponds to a term in the shared ontology. We therefore define a source ontology, an ontology that defines the meaning of specific classifications used in the source, to consist of a set of class definitions. These definitions are legal expressions

over terms from the shared ontology built using a terminological language that defines operators for the relations also defined in the shared ontology:

Definition 2.3 (source ontology). *A source ontology is a tuple* $\langle S, C, d \rangle$, *where* $S = \langle ST_S, T_S, R_S, A_S \rangle$ *is a shared ontology,* C *is a set of class names not from the set of terms in* S, L *is a terminological language and* d *is a function that assigns expressions* δ_i *to class names* C_i *in* C *such that:*

- δ_i *only refers to relations in* R_S,
- L *is defined over* T_S.

In the following we refer to δ_i *as the definition of* C_i, *which is denoted by* $d(C_i)$.

Given a source ontology we can perform terminological reasoning over the definition of classes contained therein by considering the set of axioms from the shared ontology, the definitions of relations and the set of class definitions. Together, these elements form a terminological knowledge-base that can be used by suitable description-logic reasoners in order to provide standard inference services such as classification and retrieval. How these inference services are used for retrieval and integration will be discussed in Chap. 6.

2.5 Conclusions

The use of ontologies is a straightforward and promising approach in order to explicate contextual information and to make a semantics-preserving translation possible. Especially, ontologies could be used for the specification of a source-independent shared vocabulary (domain ontology) whose concepts are used to describe the specific contextual information of different information sources to be integrated (application ontologies). The use of a common vocabulary as a basis for the context specifications is assumed to enable us to perform (semi-)automatic translations between different contexts that preserve the intended meaning of the translated terms to a large extent.

The central question is how to actually capture information semantics in ontologies. A strategy is needed that determines what kinds of ontologies are needed and how they can be built. This strategy has to trade-off globalized representations that provide a common basis for defining and comparing information semantics and local representations that capture the specific conceptual choices made in the design of individual information sources. In order to be comparable, these local definitions should be based on terms defined globally. Linguistic resources and top-level ontologies provide guidance in the choice for a global vocabulary. The representational framework defined in the first part of this book then provides operators for composing these basic terms into more complex concept definitions and to perform terminological reasoning.

State of the research

The typical information-integration system uses ontologies to explicate the contents of an information source, mainly by describing the intended meaning of table- and data-field names. For this purpose, each information source is supplemented by an ontology which resembles and extends the structure of the information source. In a typical system, integration is done at the ontology level using either a common ontology all source ontologies are related to or fixed mappings between different ontologies. The ontology language of the typical system is based on description logics, and subsumption reasoning is used in order to compute relations between different information sources and sometimes to validate the result of an integration. The process of building and using ontologies in the typical system is supported by specialized tools in terms of editors.

Open questions

The description of the typical integration system shows that reasonable results have been achieved on the technical side of using ontologies for intelligent information integration. Only the use of mappings is an exception. It seems that most approaches still use ad hoc or arbitrary mappings especially for the connection of different ontologies. There are approaches that try to provide well-founded mappings, but they either rely on assumptions that cannot always be guaranteed or they face technical problems. We conclude that there is a need to investigate mappings on a theoretical and an empirical basis.

Beside the mapping problem, we found a striking lack of sophisticated methodologies supporting the development and use of ontologies. Most systems only provide tools. If there is a methodology it often only covers the development of ontologies for a specific purpose which is prescribed by the integration system. The comparison of different approaches, however, revealed that requirements concerning ontology language and structure depend on the kind of information to be integrated and the intended use of the ontology. We therefore think that there is a need to develop a more general methodology that includes an analysis of the integration task and supports the process of defining the role of ontologies with respect to these requirements. We think that such a methodology has to be language independent, because the language should be selected based on the requirements of the application and not the other way round. A good methodology also has to cover the evaluation and verification of the decisions made with respect to language and structure of the ontology. The development of such a methodology will be a major step in the work on ontology-based information integration because it will help to integrate results already achieved on the technical side and to put these techniques to work in real-life applications.

Further reading

The first widely accepted definition of ontologies from a computer science perspective is given by Gruber [Gruber, 1993] in his seminal work. Uschold and Gruninger give excellent overview over the nature as use of ontologies [Uschold and Gruninger, 1996]. Guarino and Giaretta discuss the special character of ontologies that distinguish them from other knowledge models [Guarino and Giaretta, 1995]. We give an overview of ontology-based information integration systems in [Wache et al., 2001]. An overview of the use of ontologies at different levels of formality and extends of explication is [McGuinness, 2002].