# Ontology-Based Integration of Information — A Survey of Existing Approaches

**H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hübner**
Intelligent Systems Group, Center for Computing Technologies,
University of Bremen, P.O.B. 33 04 40, D-28334 Bremen, Germany
e-mail: {wache|vogele|visser|heiner|schuster|neumann|huebner}@tzi.de

## Abstract

We review the use on ontologies for the integration of heterogeneous information sources. Based on an in-depth evaluation of existing approaches to this problem we discuss how ontologies are used to support the integration task. We evaluate and compare the languages used to represent the ontologies and the use of mappings between ontologies as well as to connect ontologies with information sources. We also enquire into ontology engineering methods and tools used to develop ontologies for information integration. Based on the results of our analysis we summarize the state-of-the-art in ontology-based information integration and name areas of further research activities.

## 1 Motivation

The so-called information society demands complete access to available information, which is often heterogeneous and distributed. In order to establish efficient information sharing, many technical problems have to be solved. First, a suitable information source must be located that might contain data needed for a given task. Finding suitable information sources is a problem addressed in the areas of information retrieval and information filtering [Belkin and Croft, 1992]. Once the information source has been found, access to the data therein has to be provided. This means that each of the information sources found in the first step have to work together with the system that is querying the information. The problem of bringing together heterogeous and distributed computer systems is known as *interoperability problem.*

Interoperability has to be provided on a technical and on an informational level. In short, information sharing not only needs to provide full accessibility to the data, it also requires that the accessed data may be processed and interpreted by the remote system. Problems that might arise owing to heterogeneity of the data are already well-known within the distributed database systems community (e.g. [Kim and Seo, 1991], [Kashyap and Sheth, 1996a]): *structural heterogeneity* (schematic heterogeneity) and *semantic heterogeneity* (data heterogeneity) [Kim and Seo, 1991]. Structural heterogeneity means that different information systems store their data in different structures. Semantic heterogeneity considers the contents of an information item and its intended meaning.

In order to achieve semantic interoperability in a heterogeneous information system, the *meaning* of the information that is interchanged has to be understood across the systems. Semantic conflicts occur whenever two contexts do not use the same interpretation of the information. Goh identifies three main causes for semantic heterogeneity [Goh, 1997]:

- *Confounding conflicts* occur when information items seem to have the same meaning, but differ in reality, e.g. owing to different temporal contexts.
- *Scaling conflicts* occur when different reference systems are used to measure a value. Examples are different currencies.
- *Naming conflicts* occur when naming schemes of information differ significantly. A frequent phenomenon is the presence of homonyms and synonyms.

The use of ontologies for the explication of implicit and hidden knowledge is a possible approach to overcome the problem of semantic heterogeneity. Uschold and Grüninger mention interoperability as a key application of ontologies, and many ontology-based approaches [Uschold and Grüniger, 1996] to information integration in order to achieve interoperability have been developed.

In this paper we present a survey of existing solutions with special focus on the use of ontologies in these approaches. We analyzed about 25 approaches to intelligent information integration including SIMS, TSIMMIS, OBSERVER, CARNOT, Infosleuth, KRAFT, PICSEL, DWQ, Ontobroker, SHOE and others with respect to the role and use of ontologies. Most of these systems use some notion of ontologies. We only consider these approaches. A further criterion is the focus of the approach on the integration of information sources. We therefore do not consider approaches to the integration of knowledge bases. We evaluate the remaining approaches according to four main criteria:

**Use of Ontologies:** The role and the architecture of the ontologies influence heavily the representation formalism of an ontology.

**Ontology Representation:** Depending on the use of the ontology, the representation capabilities differ from approach to approach.

**Use of Mappings:** In order to support the integration process the ontologies have to be linked to actual information. If several ontologies are used in an integration system, mapping between the ontologies are also important.

**Ontology Engineering:** Before an integration of information sources can begin the appropriate ontologies have to be acquired or to be selected for reuse. How does the integration approach support the acquisition or reuse of ontologies?

In the following we discuss these points on the basis of our experiences from the comparison of different systems. Doing this we will not consider single approaches, but rather refer to typical representatives. In section 2 we discuss the use of ontologies in different approaches and common ontology architectures. The use of different representations, i.e. different ontology languages, is discussed in section 3. Mappings used to connect ontologies to information sources and inter-ontology mappings are the topic of section 4, while section 5 covers methodologies and tool-support for the ontology engineering process. We conclude with a summary of the state-of-the-art and the direction for further research in the area of ontology-based information integration.

## 2 The Role of Ontologies

Initially, ontologies are introduced as an "explicit specification of a conceptualization" [Gruber, 1993]. Therefore, ontologies can be used in an integration task to describe the semantics of the information sources and to make the contents explicit (section 2.1). With respect to the integration of data sources, they can be used for the identification and association of semantically corresponding information concepts.

However, in several projects ontologies take over additional tasks. These tasks are discussed in section 2.2.

### 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 way of how to employ the ontologies. In general, three different directions can be identified: *single ontology approaches*, *multiple ontologies approaches* and *hybrid approaches*. Figure 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.*, 2001]). 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. 1a). All information sources are related to one global ontology. A prominent approach of this kind of ontology integration is SIMS [Arens *et al.*, 1996]. SIMS model of the application domain includes a hierarchical terminological knowledge base with nodes representing objects, actions, and states. An independent model

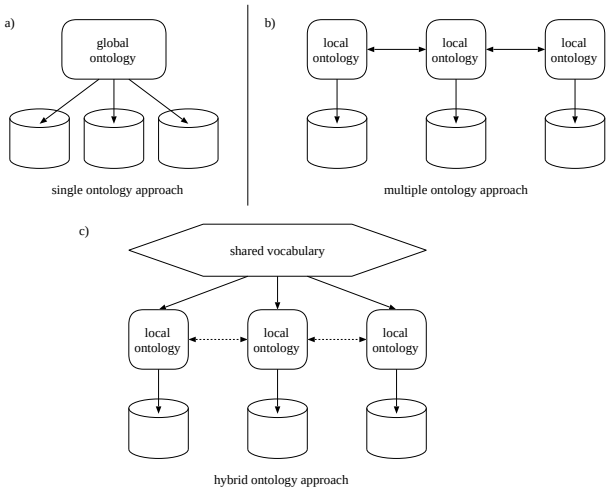


Figure 1: The three possible ways for using ontologies for content explication

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, 1993]).

Single ontology approaches can be applied to integration problems where all information sources to be integrated provide nearly the same view on a domain. But if one information source has a different view on 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. 1b). For example, in OBSERVER [Mena *et al.*, 1996] 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 can not 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 respect to other sources or their ontologies — no common ontology with the agreement of all sources are 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 inter-ontology mapping is provided (see 4.2). The inter-ontology 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 on a domain e.g. different aggregation and granularity of the ontology concepts. We believe that in practice the inter-ontology 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. 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 [Goh, 1997; Wache *et al.*, 1999]. The shared vocabulary contains basic terms (the primitives) of a domain. In order to build complex terms of a source ontologies 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 *et al.*, 2000b].

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 *et al.*, 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 number). In BUSTER [Stuckenschmidt *et al.*, 2000b], the shared vocabulary is a (general) ontology, which covers all possible refinements. E.g. the general ontology defines the attribute value ranges of its concepts. A source ontology is one (partial) refinement of the general ontology, e.g. 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, existing ontologies cannot be reused easily, but have to be re-developed from scratch, because all source ontologies have to refer to the shared vocabulary.

The following table summarizes the benefits and drawbacks of the different ontology approaches:

| | Single Ontology Approaches | Multiple Ontology Approaches | Hybrid Ontology Approaches |
|---|---|---|---|
| implementation effort | straight-forward | costly | reasonable |
| semantic heterogeneity | similar view of a domain | supports heterogen-eous views | supports heterogen-eous views |
| adding/ removing of sources | need for some adap-tion in the global ontology | providing a new source ontology; relating to other ontologies | providing a new source ontology; |
| comparing of multiple ontologies | — | difficult because of the lack of a common vocabulary | simple because ontologies use a common vocabulary |

Table 1: Benefits and drawbacks of the different ontology-based integration approaches

## 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 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 sub-queries for each appropriate source, collects and combines the query results, and returns the results.

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 where 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 sub-query is correct with respect to a global query if the local sub-query provides a part of the queried answers, i.e. the sub-queries must be contained in the global query (query containment) [Calvanese *et al.*, 2001; Goasdoué *et al.*, 1999]. 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.*, 2001] 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 sub-queries can be unfolded to their ontology concepts. The sub-queries are correct, i.e. are contained in the global query, if their ontology concepts are subsumed by the global ontology concepts. The PICSEL project [Goasdoué *et al.*, 1999] 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 can not be measured, it is impossible to make any statements about the quality of the verification.

## 3 Ontology Representations

A question that arises from the use of ontologies for different purposes in the context of information integration is about the nature of the ontologies used. Investigating this question we mainly focus on the kind of languages used and the general structures found. We do not discuss ontology contents, because we think that the contents strongly depends on the kind of information that has to be integrated. We further restrict the evaluation to an object-centered knowledge representation system which in most systems forms the core of the languages used.

The first thing we have to notice when we investigate different approaches to intelligent information integration based on ontologies is the overwhelming dominance of systems using some variants of description logics in order to represent ontologies. The most cited language is CLASSIC [Borgida *et al.*, 1989] which is used by different systems including OBSERVER [Mena *et al.*, 1996], SIMS [Arens *et al.*, 1996] and the work of Kashyap and Sheth [Kashyap and Sheth, 1996b]. Other terminological languages used are GRAIL [Rector *et al.*, 1997] (the Tambis Approach [Stevens *et al.*, 2000]), LOOM [MacGregor, 1991] and OIL [Fensel *et al.*, 2000] which is used for terminology integration in the BUSTER approach [Stuckenschmidt and Wache, 2000].

Beside the purely terminological languages mentioned above there are also approaches using extensions of description logics which include rule bases. Known uses of extended languages are in the PICSEL system using CARIN, a description logic extended with function-free horn rules [Goasdoué *et al.*, 1999] and the DWQ [Calvanese *et al.*, 2001] project. In the latter approach $\mathcal{AL} - log$ a combination of a simple description logics with Datalog is used [Donini *et al.*, 1998]. [Calvanese *et al.*, 2001] use the Logic $\mathcal{DLR}$ a description logic with n-ary relations for information integration in the same project. The integration of description logics with rule-based reasoning makes it necessary to restrict the expressive power of the terminological part of the language in order to remain decidable [Levy and Rousset, 1996].

The second main group of languages used in ontology-based information integration systems are classical frame-based representation languages. Examples for such systems are COIN [Goh, 1997], KRAFT [Preece *et al.*, 1999], Infosleuth [Woelk and Tomlinson, 1994] and Infomaster [Genesereth *et al.*, 1997]. Languages mentioned are Ontolingua [Gruber, 1993] and OKBC [Chaudhri *et al.*, 1998]. There are also approaches that directly use F-Logic [Kifer *et al.*, 1995] with a self-defined syntax (Ontobroker [Fensel *et al.*, 1998] and COIN [Goh, 1997]). For an analysis of the expressive power of these languages, we refer to Corcho and Gomez-Perez [Corcho and Gómez-Pérez, 2000] who evaluated different ontology languages including the ones mentioned above.

## 4 Use of Mappings

The task of integrating heterogeneous information sources put ontologies in context. They cannot be perceived as stand-alone models of the world but should rather be seen as the glue that puts together information of various kinds. Consequently, the relation of an ontology to its environment plays an essential role in information integration. We use the term mappings to refer to the connection of an ontology to other parts of the application system. In the following, we discuss the two most important uses of mappings required for information integration: mappings between ontologies and the information they describe and mappings between different ontologies used in a system.

### 4.1 Connection to Information Sources

The first and most obvious application of mappings is to relate the ontologies to the actual contents of an information source. Ontologies may relate to the database scheme but also to single terms used in the database. Regardless of this distinction, we can observe different general approaches used to establish a connection between ontologies and information sources. We briefly discuss these general approaches in the sequel.

**Structure Resemblance** A straightforward approach to connecting the ontology with the database scheme is to simply produce a one-to-one copy of the structure of the database and encode it in a language that makes automated reasoning

possible. The integration is then performed on the copy of the model and can easily be tracked back to the original data. This approach is implemented in the SIMS mediator [Arens *et al.*, 1996] and also by the TSIMMIS system [Chawathe *et al.*, 1994].

**Definition of Terms** In order to make the semantics of terms in a database schema clear it is not sufficient to produce a copy of the schema. There are approaches such as BUSTER [Stuckenschmidt and Wache, 2000] that use the ontology to further define terms from the database or the database scheme. These definitions do not correspond to the structure of the database, these are only linked to the information by the term that is defined. The definition itself can consist of a set of rules defining the term. However, in most cases terms are described by concept definitions.

**Structure Enrichment** is the most common approach to relating ontologies to information sources. It combines the two previously mentioned approaches. A logical model is built that resembles the structure of the information source and contains additional definitions of concepts. A detailed discussion of this kind of mapping is given in [Kashyap and Sheth, 1996a]. Systems that use structure enrichment for information integration are OBSERVER [Mena *et al.*, 1996], KRAFT [Preece *et al.*, 1999], PICSEL [Goasdoué *et al.*, 1999] and DWQ [Calvanese *et al.*, 2001]. While OBSERVER uses description logics for both structure resemblance and additional definitions, PICSEL and DWQ defines the structure of the information by (typed) horn rules. Additional definitions of concepts mentioned in these rules are done by a description logic model. KRAFT does not commit to a specific definition scheme.

**Meta-Annotation** A rather new approach is the use of meta annotations that add semantic information to an information source. This approach is becoming prominent with the need to integrate information present in the World Wide Web where annotation is a natural way of adding semantics. Approaches which are developed to be used on the World Wide Web are Ontobroker [Fensel *et al.*, 1998] and SHOE [Heflin and Hendler, 2000b]. We can further distinguish between annotations resembling parts of the real information and approaches avoiding redundancy. SHOE is an example for the former, Ontobroker for the latter case.

### 4.2 Inter-Ontology Mapping

Many of the existing information integration systems such as [Mena *et al.*, 1996] or [Preece *et al.*, 1999] use more than one ontology to describe the information. The problem of mapping different ontologies is a well known problem in knowledge engineering. We will not try to review all research that is conducted in this area. We rather discuss general approaches that are used in information integration systems.

**Defined Mappings** A common approach to the ontology mapping problem is to provide the possibility to define mappings. This approach is taken in KRAFT [Preece *et al.*, 1999], where translations between different ontologies are done by special mediator agents which can be customized to translate between different ontologies and even different languages. Different kinds of mappings are distinguished in this approach starting from simple one-to-one mappings between classes and values up to mappings between compound expressions. This approach allows a great flexibility, but it fails to ensure a preservation of semantics: the user is free to define arbitrary mappings even if they do not make sense or produce conflicts.

**Lexical Relations** An attempt to provide at least intuitive semantics for mappings between concepts in different ontologies is made in the OBSERVER system [Mena *et al.*, 1996]. The approaches extend a common description logic model by quantified inter-ontology relationships borrowed from linguistics. In OBSERVER, relationships used are *synonym, hypernym, hyponym, overlap, covering* and *disjoint*. While these relations are similar to constructs used in description logics they do not have a formal semantics. Consequently, the subsumption algorithm is rather heuristic than formally grounded.

**Top-Level Grounding** In order to avoid a loss of semantics, one has to stay inside the formal representation language when defining mappings between different ontologies (e.g. DWQ [Calvanese *et al.*, 2001]). A straightforward way to stay inside the formalism is to relate all ontologies used to a single top-level ontology. This can be done by inheriting concepts from a common top-level ontology. This approach can be used to resolve conflicts and ambiguities (compare [Heflin and Hendler, 2000b]). While this approach allows to establish connections between concepts from different ontologies in terms of common superclasses, it does not establish a direct correspondence. This might lead to problems when exact matches are required.

**Semantic Correspondences** An approach that tries to overcome the ambiguity that arises from an indirect mapping of concepts via a top-level grounding is the attempt to identify well-founded semantic correspondences between concepts from different ontologies. In order to avoid arbitrary mappings between concepts, these approaches have to rely on a common vocabulary for defining concepts across different ontologies. Wache [1999] uses semantic labels in order to compute correspondences between database fields. Stuckenschmidt et. al. build a description logic model of terms from different information sources and shows that subsumption reasoning can be used to establish relations between different terminologies. Approaches using formal concept analysis (see above) also fall into this category, because they define concepts on the basis of a common vocabulary to compute a common concept lattice.

## 5 Ontological Engineering

The previous sections provided information about the use and importance of ontologies. Hence, it is crucial to support the

development process of ontologies. In this section, we will describe how the systems provide support for the ontological engineering process. This section is divided into three subsections: In the first subsection we give a brief overview of development methodology. The second subsection is an overview of supporting tools and the last subsection describes what happens when ontologies change.

### 5.1 Development Methodology

Lately, several publications about ontological developments have been published. Jones et al. [1998] provide an excellent but short overview of existing approaches (e.g. METHONTODOLOGY [Gómez-Pérez, 1998] or TOVE [Fox and Grüninger, 1998]). Uschold and Grüninger [1996] and Gómez-Pérez et al. [1996] propose methods with phases that are independent of the domain of the ontology. These methods are of good standards and can be used for comparisons. In this section, we focus on the proposed method from Uschold and Grüninger as a 'thread' and discuss how the integrated systems evaluated in this paper are related to this approach.

Uschold and Grüninger defined four main phases:

1. Identifying a purpose and scope: Specialization, intended use, scenarios, set of terms including characteristics and granularity
2. Building the ontology
   (a) Ontology capture: Knowledge acquisition, a phase interacting with requirements of phase 1.
   (b) Ontology coding: Structuring of the domain knowledge in a conceptual model.
   (c) Integrating existing ontologies: Reuse of existing ontologies to speed up the development process of ontologies in the future.
3. Evaluation: Verification and Validation.
4. Guidelines for each phase.

In the following paragraphs we describe integration systems and their methods for building an ontology. Further, we discuss systems without an explicit method where the user is only provided with information in the direction in question. The second type of systems can be distinguished from others without any information about a methodology. This is due to the fact that they assume that ontologies already exist.

**Infosleuth:** This system semi-automatically constructs ontologies from textual databases [Hwang, 1999]. The methodology is as follows: first, human experts provide a small number of *seed words* to represent high-level concepts. This can be seen as the identification of purpose and scope (phase 1). The system then processes the incoming documents, extracting phrases that involve seed words, generates corresponding concept terms, and then classifies them into the ontology. This can be seen as ontology capturing and part of coding (phases 2a and 2b). During this process the system also collects seed word-candidates for the next round of processing. This iteration can be completed for a predefined number of rounds. A human expert verifies the classification after each round (phase 3). As more documents arrive, the ontology expands and the expert is confronted with the new concepts. This is a significant feature of this system. Hwang calls this 'discover-and-alert' and indicates that this is a new feature of his methodology. This method is conceptually simple and allows effective implementation. Prototype implementations have also shown that the method works well. However, problems arise within the classification of concepts and distinguishing between concepts and non-concepts.

Infosleuth requires an expert for the evaluation process. When we consider that experts are rare and their time is costly this procedure is too expert-dependent. Furthermore, the integration of existing ontologies is not mentioned. However, an automatic verification of this model by a reasoner would be worthwhile considering.

**KRAFT:** offers two methods for building ontologies: the building of shared ontologies [Jones, 1998] and extracting of source ontologies [Pazzaglia and Embury, 1998].

**Shared ontologies:** The steps of the development of shared ontologies are *(a) ontology scoping, (b) domain analysis, (c) ontology formalization, (d) top-level-ontology*. The minimal scope is a set of terms that is necessary to support the communication within the KRAFT network. The domain analysis is based on the idea that changes within ontologies are inevitable and the means to handle changes should be provided. The authors pursue a domain-led strategy [Paton *et al.*, 1991], where the shared ontology fully characterizes the area of knowledge in which the problem is situated. Within the ontology formalization phase the fully characterized knowledge is defined formally in classes, relations and functions. The top-level-ontology is needed to introduce predefined terms/primitives.

If we compare this to the method of Uschold and Grüninger we can conclude that ontology scoping is weakly linked to phase 1. It appears that ontology scoping is a set of terms fundamental for the communication within the network and therefore can be seen as a vocabulary. On the other hand, the authors say that this is a *minimal* set of terms which implies that more terms exist. The domain analysis refers to phases 1 and 2a whereas the ontology formalization refers to phase 2b. Existing ontologies are not considered.

**Extracting ontologies:** Pazzaglia and Embury [1998] introduce a bottom-up approach to extract an ontology from existing shared ontologies. This extraction process consists of two steps. The first step is a syntactic translation from the KRAFT exportable view (in a native language) of the resource into the KRAFT-schema. The second step is the ontological upgrade, a semi-automatic translation plus knowledge-based enhancement, where local ontology adds knowledge and further relationships between the entities in the translated schema.

This approach can be compared to phase 2c, the integration of existing ontologies. In general, the KRAFT methodology lacks the evaluation of ontologies and the general purpose scope.

**Ontobroker:** The authors provide information about phase 2, especially 2a and 2b. They distinguish between three

classes of web information sources (see also [Ashish and Knoblock, 1997]): *(a) Multiple-instance sources* with the same structure but different contents, *(b) single-instance sources* with large amount of data in a structured format, and *(c) loosely structured pages* with little or no structure. Ontobroker [Decker *et al.*, 1999] has two ways of formalizing knowledge (this refers to phase 2b). First, sources from (a) and (b) allow to implement wrappers that automatically extract factual knowledge from these sources. Second, sources with little or no knowledge have to be formalized manually. A supporting tool called OntoEdit [Staab *et al.*, 2000] is an ontology editor embedded in the ontology server and can help to annotate the knowledge. OntoEdit is described later in this section.

Apart from the connection to phase 2 the Ontobroker system provides no information about the scope, the integration of existing ontologies, or the evaluation.

**SIMS:** An independent model of each information source must be described for this system, along with a domain model that must be be defined to describe objects and actions [Arens *et al.*, 1993]. SIMS model of the application domain includes a hierarchical terminological knowledge base with nodes representing objects, actions, and states. In addition, it includes indications of all relationships between the nodes. Further, the authors address the scalability and maintenance problems when a new information source is added or the domain knowledge changes. As every information source is independent and modeled separately, the addition of a new source should be relatively straightforward. A graphical LOOM knowledge base builder (LOOM-KB) can be used to support this process. The domain model would have to be enlarged to accommodate new information sources or simply new knowledge (see also [MacGregor, 1990], [MacGregor, 1988]).

The SIMS model has no concrete methodology for building ontologies. However, we see links referring to phase 2a ontology capture (description of the independent model of information sources) and 2b ontology coding (LOOM-KB). The integration of existing ontologies and an evaluation phase are not mentioned.

All the other systems discussed, such as Picsel, Observer, the approach from Kayshap & Sheth, BUSTER and COIN either have no methods or do not discuss them to create ontologies. After reading papers about these various systems it becomes obvious that there is a lack of a 'real' methodology for the development of ontologies. We believe that the systematic development of the ontology is extremely important and therefore the tools supporting this process become even more significant.

## 5.2 Supporting tools

Some of the systems we discussed in this paper provide support with the annotation process of sources. This process is mainly a semantic enrichment of the information therein. In the following, we sketch the currently available tools.

- OntoEdit: This tool makes it possible to inspect, browse, codify and modify ontologies and to use these features to support the ontology development and maintenance task [Staab and Mädche, 2000]. Currently, OntoEdit supports the representation languages *(a) F-Logic including an inference engine*, *(b) OIL*, *(c) Karlsruhe RDF(S)extension*, and *(d) internal XML-based serialization of the ontology model using OXML*.
- SHOE's Knowledge Annotator: With the help of this tool, the user can describe the contents of a web page [Heflin and Hendler, 2000b]. The Knowledge Annotator has an interface which displays instances, ontologies, and claims (documents collected). The tool also provides integrity checks. With a second tool called Exposé the annotated web pages can be parsed and the contents will be stored in a repository. This SHOE-knowledge is then stored in a Parka knowledge base [Stoffel *et al.*, 1997].
- DWQ: Further development within the DWQ project leads to a tool called i·com [Franconi and Ng, 2000]. i·com is a supporting tool for the conceptual design phase. This tool uses an extended entity relationship conceptual (EER) data model and enriches it with aggregations and inter-schema constraints. i·com does not provide a methodology nor is it an annotation tool, it serves mainly for intelligent conceptual modelling.

Annotation tools such as OntoEdit and the Knowledge Annotator are relatively new on the market. Therefore, comprehensive tests to give a good evaluation have yet to be done. However, we did the first steps with OntoEdit and came to the conclusion that OntoEdit seems to be a powerful tool and worthwhile considering. This is especially true when using an integration system which does not support the development process of an ontology. Also, OntoEdit allows to verify an ontology. Tests with the Knowledge Annotator have yet to be done.

## 5.3 Ontology Evolution

Almost every author describes the evolution of an ontology as a very important task. An integration system — and the ontologies — must support adding and/or removing sources and must be robust to changes in the information source. However, integration systems which take this into account are rare. To our knowledge, SHOE is the only system that accomplishes this to-date.

**SHOE:** Once the SHOE-annotated web pages are uploaded on the web, the Exposé tool has the task to update the repositories with the knowledge from these pages. This includes a list of pages to be visited and an identification of all hypertext links, category instances, and relation arguments within the page. The tool then stores the new information in the PARKA knowledge base. Heflin and Hendler [2000a] analyzed the problems associated with managing dynamic ontologies through the web. By adding revision marks to the ontology, changes and revision become possible. The authors illustrated that revisions which add categories and relations will have no effect, and that revisions which modify rules may change the answers to queries. When categories and relations are removed, answers to queries may be eliminated.

In summary, most of the authors mention the importance of a method for building ontologies. However, only few systems really support the user with a genuine method. Infosleuth is the only system which fulfills the requirements of a methodology. However, the majority of the systems only provide support of the formalization phase (please refer to phases 2a and 2b). KRAFT, SIMS, DWQ, and SHOE are representatives of this group. The remaining systems do not include a methodology. Some systems offer some support for the annotation of information sources (e.g. SHOE). Other systems provide supporting tools for parts of ontology engineering (e.g. DWQ/i·com, OntoEdit). Only the SHOE system may be considered as a system which takes ontology evolution into account.

## 6 Summary

In this paper we presented the results of an analysis of existing information integration systems from an ontology point of view. The analysis was focused on systems and approaches with ontologies as a main element. Important questions covered in the analysis are:

**Role of the ontology:** What is the purpose of the ontology and how does it relate to other parts of the systems?

**Ontology Representation:** What are the features (expressiveness, reasoning capabilities) of the language used to represent the ontology?

**Use of Mappings:** How is the connection of an ontology to other parts of the system especially data-repositories and other ontologies implemented?

**Ontology Engineering:** Does the approach contain a methodology and tools that support the development and the use of the ontology?

We evaluated different approaches with respect to these questions. At this point, we try to summarize the lessons learned from the analysis by drawing a rough picture of the state-of-the-art implied by the systems we analyzed. On the other hand, we try to infer open problems and to define research questions that have been put forward but reqire further investigation.

**State of the Research**

We tried to illustrate the state of the art by describing a 'typical' information integration system that uses well-established technologies: The typical information integration system uses ontologies to explicate the contents of an information source, mainly by describing the intended meaning of table and datafield 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 depends 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.

## References


[Arens *et al.*, 1993] Yigal Arens, Chin Y. Chee, Chun-Nan Hsu, and Craig A. Knoblock. Retrieving and integrating data from multiple information sources. *International Journal of Intelligent and Cooperative Information Systems*, 2(2):127–158, 1993.

[Arens *et al.*, 1996] Yigal Arens, Chun-Nan Hsu, and Craig A. Knoblock. Query processing in the sims information mediator. In *Advanced Planning Technology*. AAAI Press, California, USA, 1996.

[Ashish and Knoblock, 1997] Naveen Ashish and Craig A. Knoblock. Semi-automatic wrapper generation for internet information sources. In *Second IFCIS International Conference on Cooperative Information Systems*, Kiawah Island, SC, 1997.

[Belkin and Croft, 1992] N.J. Belkin and B.W. Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, December 1992.

[Borgida *et al.*, 1989] A. Borgida, Brachman a R. J., D. L. McGuiness, and L. A. Resnick. Classic: A structural data

model for objects. In *ACM SIGMOID International Conference on Management of Data*, Portland, Oregon, USA, 1989.

[Calvanese *et al.*, 2001] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Description logics for information integration. In *Computational Logic: From Logic Programming into the Future (In honour of Bob Kowalski)*, Lecture Notes in Computer Science. Springer-Verlag, 2001. To appear.

[Chaudhri *et al.*, 1998] Vinay K. Chaudhri, Adam Farquhar, Richard Fikes, Peter D. Karp, and James P. Rice. Open knowledge base connectivity (okbc) specification document 2.0.3. Technical report, SRI International and Stanford University (KSL), April 1998.

[Chawathe *et al.*, 1994] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. The tsimmis project: Integration of heterogeneous information sources. In *Conference of the Information Processing Society Japan*, pages 7–18, 1994.

[Corcho and Gómez-Pérez, 2000] Oscar Corcho and Asuncion Gómez-Pérez. Evaluating knowledge representation and reasoning capabilities of ontology specification languages. In *Proceedings of the ECAI 2000 Workshop on Applications of Ontologies and Problem-Solving Methods*, Berlin, 2000.

[Decker *et al.*, 1999] Stefan Decker, Michael Erdmann, Dieter Fensel, and Rudi Studer. Ontobroker: Ontology based access to distributed and semi-structured information. In R. Meersman et al., editor, *Semantic Issues in Multimedia Systems. Proceedings of DS-8*, pages 351–369. Kluwer Academic Publisher, Boston, 1999.

[Donini *et al.*, 1998] F. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Al-log: Integrating datalog and description logics. *Journal of Intelligent Information Systems (JIIS)*, 27(1), 1998.

[Fensel *et al.*, 1998] Dieter Fensel, Stefan Decker, M. Erdmann, and Rudi Studer. Ontobroker: The very high idea. In *11. International Flairs Conference (FLAIRS-98)*, Sanibal Island, USA, 1998.

[Fensel *et al.*, 2000] D. Fensel, I. Horrocks, F. Van Harmelen, S. Decker, M. Erdmann, and M. Klein. Oil in a nutshell. In *12th International Conference on Knowledge Engineering and Knowledge Management EKAW 2000*, Juanles-Pins, France, 2000.

[Fox and Grüninger, 1998] Mark S. Fox and Michael Grüninger. Enterprise modelling, fall 1998, pp. 109-121. *AI Magazine*, 19(3):109–121, 1998.

[Franconi and Ng, 2000] Enrico Franconi and Gary Ng. The i.com tool for intelligent conceptual modelling. In *7th Intl. Workshop on Knowledge Representation meets Databases (KRDB'00)*, Berlin, Germany, August 2000, 2000.

[Genesereth *et al.*, 1997] Michael R. Genesereth, Arthur M. Keller, and Oliver Duschka. Infomaster: An information integration system. In *1997 ACM SIGMOD Conference.*, 1997.

[Gómez-Pérez *et al.*, 1996] Ascuncion Gómez-Pérez, M. Fernández, and A. de Vicente. Towards a method to conceptualize domain ontologies. In *Workshop on Ontological Engineering, ECAI '96*, pages 41–52, Budapest, Hungary, 1996.

[Gómez-Pérez, 1998] A. Gómez-Pérez. Knowledge sharing and reuse. In Liebowitz, editor, *The handbook on Applied Expert Systems*. ED CRC Press, 1998.

[Goasdoué *et al.*, 1999] François Goasdoué, Véronique Lattes, and Marie-Christine Rousset. The use of carin language and algorithms for information integration: The picsel project,. *International Journal of Cooperative Information Systems (IJCIS)*, 9(4):383 – 401, 1999.

[Goh, 1997] Cheng Hian Goh. *Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources*. Phd, MIT, 1997.

[Gruber, 1993] Tom Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[Gruber, 1995] Tom Gruber. Toward principles for the design of ontologies used for knowledge sharing, 1995.

[Heflin and Hendler, 2000a] Jeff Heflin and James Hendler. Dynamic ontologies on the web. In *Proceedings of American Association for Artificial Intelligence Conference (AAAI-2000)*, Menlo Park, CA, 2000. AAAI Press.

[Heflin and Hendler, 2000b] Jeff Heflin and James Hendler. Semantic interoperability on the web. In *Extreme Markup Languages 2000*, 2000.

[Hwang, 1999] Chung Hee Hwang. Incompletely and imprecisely speaking: Using dynamic ontologies for representing and retrieving information. Technical, Microelectronics and Computer Technology Corporation (MCC), June 1999.

[Jones *et al.*, 1998] D. M. Jones, T.J.M. Bench-Capon, and P.R.S. Visser. Methodologies for ontology development. In *Proc. IT&KNOWS Conference of the 15th IFIP World Computer Congress*, Budapest, 1998. Chapman-Hall.

[Jones, 1998] D.M. Jones. Developing shared ontologies in multi agent systems. Tutorial, 1998.

[Kashyap and Sheth, 1996a] V. Kashyap and A. Sheth. Schematic and semantic semilarities between database objects: A context-based approach. *The International Journal on Very Large Data Bases*, 5(4):276–304, 1996.

[Kashyap and Sheth, 1996b] Vipul Kashyap and Amit Sheth. Semantic heterogeneity in global information systems: The role of metadata, context and ontologies. In M. Papazoglou and G. Schlageter, editors, *Cooperative Information Systems: Current Trends and Applications*. 1996.

[Kifer *et al.*, 1995] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based systems. *Journal of the ACM*, 1995.

[Kim and Seo, 1991] Won Kim and Jungyun Seo. Classifying schematic and data heterogeinity in multidatabase systems. *IEEE Computer*, 24(12):12–18, 1991. problem classification of semantic heterogeneity.

[Levy and Rousset, 1996] Alon Y. Levy and Marie-Christine Rousset. Carin: A representation language combining horn rules and description logics. In *Proceedings of the 12th European Conf. on Artificial Intelligence (ECAI-96)*, pages 323–327, 1996.

[MacGregor, 1988] Robert M. MacGregor. A deductive pattern matcher. In *Seventh National Conference on Artificial Intelligence, (AAAI 88)*, pages 403–408, 1988.

[MacGregor, 1990] Robert MacGregor. The evolving technology of classification-based knowledge representation systems. In John Sowa, editor, *Principles of Semantic Networks: Explorations in the Representation of Knowledge*. Morgan Kaufman, 1990.

[MacGregor, 1991] Robert M. MacGregor. Using a description classifier to enhance deductive inference. In *Proceedings Seventh IEEE Conference on AI Applications*, pages 141–147, 1991.

[Mena *et al.*, 1996] E. Mena, V. Kashyap, A. Sheth, and A. Illarramendi. Observer: An approach for query processing in global information systems based on interoperability between pre-existing ontologies. In *Proceedings 1st IFCIS International Conference on Cooperative Information Systems (CoopIS '96)*. Brussels, 1996.

[Paton *et al.*, 1991] R.C Paton, H.S. Nwana, M.J.R. Shave, T.J.M. Bench-Capon, and S. Hughes. Foundations of a structured approach to characterising domain knowledge. *Cognitive Systems*, 3(2):139–161, 1991.

[Pazzaglia and Embury, 1998] J-C.R. Pazzaglia and S.M. Embury. Bottom-up integration of ontologies in a database context. In *KRDB'98 Workshop on Innovative Application Programming and Query Interfaces*, Seattle, WA, USA, 1998.

[Preece *et al.*, 1999] A.D. Preece, K.-J. Hui, W.A. Gray, P. Marti, T.J.M. Bench-Capon, D.M. Jones, and Z. Cui. The kraft architecture for knowledge fusion and transformation. In *Proceedings of the 19th SGES International Conference on Knowledge-Based Systems and Applied Artificial Intelligence (ES'99)*. Springer, 1999.

[Rector *et al.*, 1997] A.L. Rector, S. Bechofer, C.A. Goble, I. Horrocks, W.A. Nowlan, and W.D. Solomon. The grail concept modelling language for medical terminology. *Artificial Intelligence in Medicine*, 9:139 – 171, 1997.

[Staab and Mädche, 2000] S. Staab and A. Mädche. Ontology engineering beyond the modeling of concepts and relations. In *ECAI'2000 Workshop on on Applications of Ontologies and Problem-Solving Methods*, Berlin, 2000.

[Staab *et al.*, 2000] S. Staab, M. Erdmann, and A. Mädche. An extensible approach for modeling ontologies in rdf(s). In *First ECDL'2000 Semantic Web Workshop*, Lisbon, Portugal, 2000.

[Stevens *et al.*, 2000] R. Stevens, P. Baker, S. Bechhofer, G. Ng, A. Jacoby, N.W. Paton, C.A. Goble, and A. Brass. Tambis: Transparent access to multiple bioinformatics information sources. *Bioinformatics*, 16(2):184–186, 2000.

[Stoffel *et al.*, 1997] Kilian Stoffel, Merwyn Taylor, and James Hendler. Efficient management of very large ontologies. In *American Association for Artificial Intelligence Conference (AAAI-97)*, pages 442–447, Menlo Park, CA, 1997. AAAI/MIT Press.

[Stuckenschmidt and Wache, 2000] Heiner Stuckenschmidt and Holger Wache. Context modelling and transformation for semantic interoperability. In *Knowledge Representation Meets Databases (KRDB 2000)*. 2000.

[Stuckenschmidt *et al.*, 2000a] H. Stuckenschmidt, Frank van Harmelen, Dieter Fensel, Michel Klein, and Ian Horrocks. Catalogue integration: A case study in ontology-based semantic translation. Technical Report IR-474, Computer Science Department, Vrije Universiteit Amsterdam, 2000.

[Stuckenschmidt *et al.*, 2000b] Heiner Stuckenschmidt, Holger Wache, Thomas Vögele, and Ubbo Visser. Enabling technologies for interoperability. In Ubbo Visser and Hardy Pundt, editors, *Workshop on the 14th International Symposium of Computer Science for Environmental Protection*, pages 35–46, Bonn, Germany, 2000. TZI, University of Bremen.

[Uschold and Grüniger, 1996] M. Uschold and M. Grüniger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.

[Uschold and Grüninger, 1996] Mike Uschold and Michael Grüninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.

[Wache *et al.*, 1999] H. Wache, Th. Scholz, H. Stieghahn, and B. König-Ries. An integration method for the specification of rule–oriented mediators. In Yahiko Kambayashi and Hiroki Takakura, editors, *Proceedings of the International Symposium on Database Applications in Non-Traditional Environments (DANTE'99)*, pages 109–112, Kyoto, Japan, November, 28-30 1999.

[Wache, 1999] Holger Wache. Towards rule-based context transformation in mediators. In S. Conrad, W. Hasselbring, and G. Saake, editors, *International Workshop on Engineering Federated Information Systems (EFIS 99)*, Kühlungsborn, Germany, 1999. Infix-Verlag.

[Woelk and Tomlinson, 1994] Darrell Woelk and Christine Tomlinson. The infosleuth project:intelligent search management via semantic agents. In *Second World Wide Web Conference '94: Mosaic and the Web*, 1994.