NoE InterOp

WP8, subtask 3

State of the Art Report

Ontology Interoperability

-Draft version 0.3.2-

Introduction

In the last period, there has been much research related to the new fronteer of the World Wide Web - the so called Semantic Web . The hope is that the Semantic Web can reduce, and in some cases solve, some of the problems encountered with the current web, where the resources (such as documents, web pages, etc.) contain data expressed in a machine-readable, but not machine-understandable form. The goal is to enable computers to process the interchanged data in a more "intelligent" way. To this end ontologies are seen as the enabling technology, that allows the formalization of the semantics of information and the unambiguous interpretation.

Ontologies play a prominent role on the Semantic Web; they could be key elements in many applications such as information retrieval, web-services search and composition, and web site management (for organization and navigation). Researchers do also believe that ontologies will contribute to solve the problem of interoperability between software applications across different organisations, providing a shared understanding of common domains. Ontologies allow applications to agree on the terms that they use when communicating. Thus, ontologies, if shared among the interoperating applications, allow the exchange of data to take place not only at a syntactic level, but also at a semantic level.

However, the Semantic Web community agrees on the fact that a single, universal ontology can not be built. Because of the large variety of information sources on the Web, documents on it will inevitably result from many different ontologies. We foresee that in the future most information systems will use ontologies. We must expect an explosion in the number of ontologies, even when considering similar domains.

For these reasons, a key challenge in building the Semantic Web is enabling the interoperability among different ontologies. Ontologies can interoperate only if correspondences between their elements have been identified and established. Today, if two ontologies need to interoperate, the mapping is mainly achieved by hand. But this task is is tedious, error-prone, and time consuming. According to the above scenario, the manual solution of the ontology interoperability problem could be a bottleneck in building a network of cooperating information management systems. Hence, the introduction of new methodologies and user-friendly tools that support the knowledge engineer in discovering semantic correspondences is crucial to the success of the Semantic Web.

In this chapter we aim at surveying the existing approaches to establishing correspondences between different ontologies.

Another issue, strictly related to ontology interoperability, is the development of creation and maintenance environments that support evolution and versioning of ontologies as they become larger, more distributed, and longer-lived; mappings between two versions of a same ontology can put in light the changes occurred.

In this chapter we will first describe which kind of mismatches can occur among different ontologies and then will see which are the solutions available in literature, to deal with such mismatches.

1.1 Preliminary definitions

When talking about ontology interoperability the following, in literature, are considered relevant operations:

- Ontology mapping/matching
- Ontology alignment
- Ontology translation
- Ontology transformation
- Ontology merging/integrating
- Ontology checking
- Ontology evolution/versioning
- Mappings management

In the next sections we will describe in more detail each of them.

1.1.1 Ontology mapping/matching

Establishing mappings between two ontologies means that for each entity (concept, relation, attribute,etc) in one ontology we try to find a corresponding entity in the second ontology, with the same or the closest intended meaning; usually this correspondance is expressed by 1 to 1 functions. We will see in the next section that mappings can be established after an analysis of the **similarity**, according to a certain metric, of the entities in he compared ontologies. It is important to note that the mapping process do not modify the involved ontologies and produce, as output, only a set of corespondences. See also [KS03] [OW03] for surveys on ontology mapping methods.

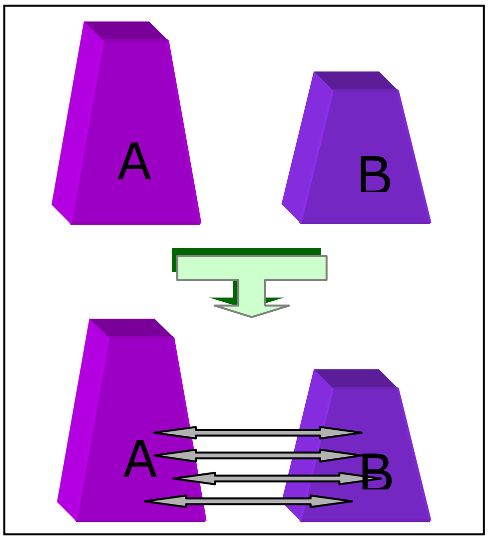

Figure 1: Ontology mapping

1.1.2 Ontology alignment

In literature, ontology alignment is considered a synonym of ontology mapping. In this chapter we will refer to ontology alignment as the process of bringing two or more ontologies into mutual agreement, making them consistent and coherent with one and another; this process may require a transformation of the involved ontologies eliminating the "non-needed" information (wrt a negotiation of the interoperability needs) while the missing information must be integrated. In contrast with mapping, this operation might result in changes of one or more of the involved ontologies [KS03][KY02].

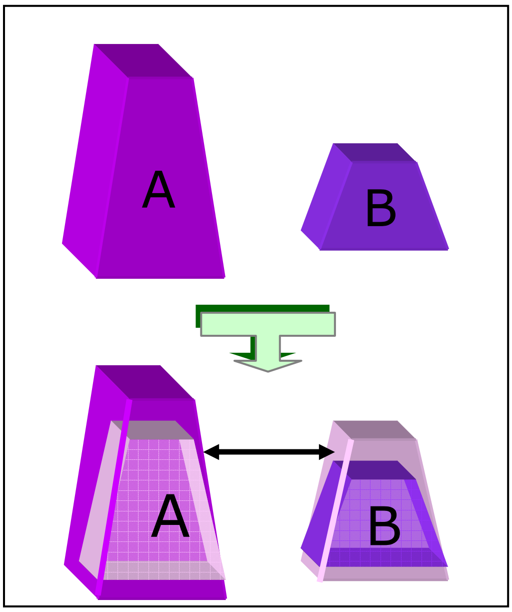

Figure 2: Ontology alignment

1.1.3 Ontology translation

Sometimes there's the need to change the formalism in which a particular ontology is expressed, for example if we decide to reuse an ontology (or part of an ontology) using a tool or language that is different from those ones in which the ontology is available; a good translation will leave the semantics of the translated ontology unaltered, or as closest as possible, to the original [refs].

1.1.4 Ontology transformation

This process consist in changing the structure of an ontology leaving unaltered its semantics (lossless transformations) or modifying it slightly (if we have a loss of information we talk about "lossy" transformations) to make it suitable for different purposes other than the original one.

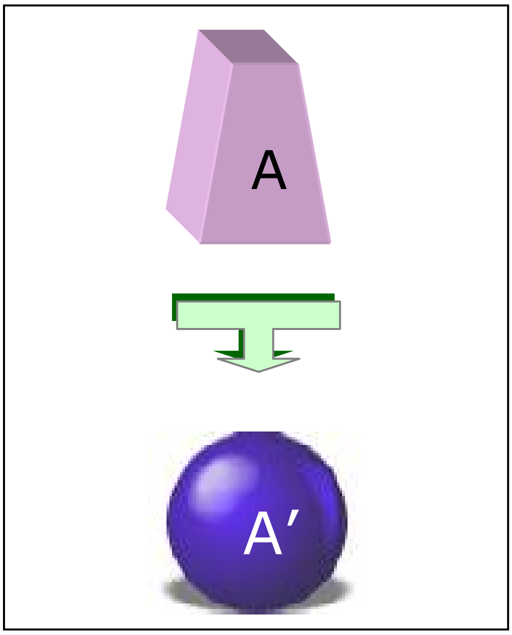

Figure 3: Ontology translation/tansformation

1.1.5 Ontology *merging/integrating*

When we build a new ontology starting from two or more existing ontologies (in general with overlapping parts) we talk about *ontology merging* while when reusing existing ontologies, assembling extending and specializing them, this is usually referred as *ontology integration* [KS03].

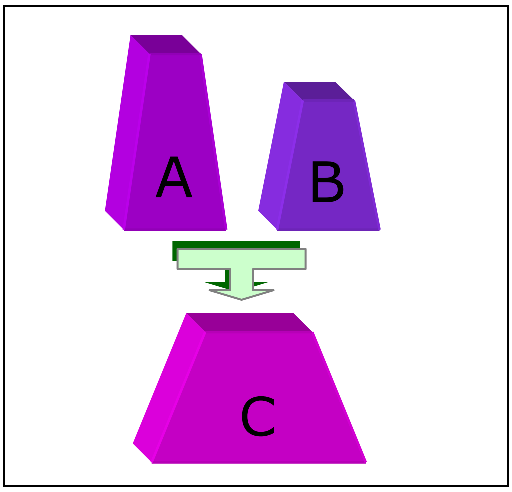

Figure 4: Ontology merging

1.1.6 Ontology checking

Once an ontology has been created (as the result of an integration/merging) or transformed (as the result of a translation/transformation/alignment), the result must be checked in order to identify possibly inconsistencies or information losses. Validation of an ontologies is commonly performed by reasoners.

1.1.7 Ontology evolution/ versioning

Ontologies describe a reusable portion of knowledge about a specific domain. However, they may change over time. Domain changes, adaptations to different tasks, or changes in the conceptualization require modifications of the ontology. A versioning mechanism may be useful to reduce the problems related to ontology changes. Fort example if we have a web page annotated with an expression over the elements of an ontology and, successively such ontology changes, the annotations could contain unknown terms or have a different interpretation [NK03].

1.1.8 Mappings management

Mappings correlate entities in different ontologies; they can be expressed in the same formalism as the ontologies, as well as into another "ad hoc" formalism. Once mappings are created, important needs arise in the context of their management:

- Are mappings resilient to ontology changes over time?

If the source ontologies are modified or transformed, existing mappings relating such ontologies, could become not valid, so there's a need for a mechanism to update them.

- Is there any mechanism for composition of mappings?

Consider you have the ontologies O_1, O_2,O_3, and that M_1 is the set of mappings established between O_1 and O_2 and M_2 is the set of mappings established between O_2 and O_3. An algebra for the composition of mappings should allow the creation of the set of the set M_3, that contains the mappings that correlate elements between O_1 and O_3.

- Is there any kind of second order mapping?

Mappings themselves could be examined to discover interrelation between them; a possible scenario in which this "second order" mappings could be used is the following: given the ontologies O_1, O_2, P_1, P_2 and the set of mappings M_1 correlating O_1 and O_2 and M_2 correlating P_1 and P_2, and the set *M* correlating M_1 and M_2, we could be able to automatically create mappings between O_1 and P_1, O_2 and P_2, etc.

1.2 Semantic similarity among ontologies

In order to detect correspondances between different ontologies, describing overlapping domains, the **semantic similarity** between every in them, according to a specified metric must be computed; this computation associate to each pair of entities a coefficient (usually in the normailzed interval [0,1]) proportional to how much close they are; this cofficient is usally referred as the **similarity degree** of the entities. Obviously several definitions of similarity are possible, each being appropriate for given situations. Similarity analysis can be carried out at the intensional level, inspecting entities descriptions to find structural similarity, or analyizing the terminological part of the descriptions (e.g. the label used to name a concept or a relation) and so on. A difference in the way one entity is represented in two different conceptualisation is usually referred as an **ontology mismatch**, that will be described in detail in the next section. At the extensional level a typical similarity measure (see section about GLUE, FCA-Merge) is the notion of the joint probability distribution between any two concepts A and B (i.e. the fraction of instances that belong both to A and B).

1.3 Ontology mismatches

In general we can say that mismatches between concept descriptions in two ontologies can occur at three different levels: **language, content, organisation**. In presenting this classification we referred to the classification made by Klein [KL01], modifying it slightly.

1.3.1 Language level mismatches

These mismatches are related to the language or the representation formalism used to represent the ontologies to be compared.

Syntax

Different ontology modelling languages, usually differ in syntax; for example, to define the class of cars in OWL, the used syntax is *<owl:Class rdf:ID = "Car">* whereas in LOOM, you have to use the expression *(defconcept Car)* to define the same class. This mismatch usually is one of the less difficult to solve using some mechanism of language translation. Several tools in literature, as a first step, transform the input sources to a common ontology language (see section about Onion, Prompt).

Expressive power

The possibility of translating one ontology in a given language to another is strictly related to the expressive power of the source and target languages; a difference in the expressive power means that one of the language is able to express things not expressible in the other and, consequently that not every ontology of one must be translated in the other, and/or vice versa, without loss of information. For example KIF's [ref] expressive power is that of full first order logic while with OWL [ref] DL it is possible to express only a subset of it.

Semantics of primitives

A more subtle problem can be detected when two languages contain constructs with the same name but the semantics of such constructs differ. For example in OIL RDF Schema [FH01] the interpretation of the *<rdfs:domain>* construct is the intersection of the arguments whereas in RDF-S [ref] is the union of the arguments.

1.3.2 Organisation level mismatches

Organisation level mismatches can be found when two or more ontologies that describe (totally or partially) overlapping domains are compared and there is a difference in the way the conceptualization is specified. Obviously these mismatches may occur when the ontologies are written in the same language, as well as when the languages are different.

Synonym terms/ Multi-Language

Concepts with the same intended meaning in different ontologies are labelled with different names. For example is the use of term "car" in one ontology and the term "automobile" in another ontology. A special kind is the use of different languages to name the concepts (the name of the concepts in one ontology are in Italian whereas in the another are in French)

Homonym terms

The compared ontologies contains overlapping terminology but with different meanings. For example, the term "conductor" in the music domain and in the electric engineering domain have completely different meaning.

Concept structuring

This type of differences may depend on the design decisions of the knowledge engineer i.e. several choices can be made for the modelling of concepts in the ontologies. An example is the way of represent the concept Person: in one ontology we have the definition of a class "Person" with two disjoint subclasses "Man", "Woman" while in another we have a class "Person" with a qualifying attribute "Sex".

In other cases the same thing can be expressed using different language constructs. For example, in some languages, it is possible to state explicitly that two classes are disjoint (e.g. disjoint A B), whereas it is necessary to use negation in subclass statements in other languages (e.g.. A subclass-of (NOT B), B subclass-of (Not A)). Some languages have specific constructs to state classes equivalence whereas others may express the same using subclass constructs.

Encoding

An encoding mismatch is a difference in value formats, like expressing temperature in Celsius or Fahrenheit degrees. Usually this mismatch can be solved using conversion functions.

Paradigm

Different paradigms can be used to represent concepts such as time, action, plans, causality, propositional attitudes, etc. For example, one model might use temporal representations based on interval logic while another might use a representation based on point.

1.3.3 Content level mismatches

Scope

Scope mismatches happen when two concepts seem to have the same intended meaning, but do not have the same instances, although they may intersect. A typical example is the class "employee", where several administrations use slightly different concepts of employee, (see Wiederhold [MW00]).

Model coverage and granularity

This is a mismatch in the part of the domain that is covered by the ontology, or the level of detail to which that domain is modelled. In Chalupsky [C00] we find the example of an ontology about cars: one ontology might model cars but not trucks. Another one might represent trucks but only classify them into a few categories, while a third ontology might make very fine-grained distinctions between types of trucks based on their physical structure, weight, purpose, etc. and a fourth ontology doesn't model cars and trucks at all.

1.4 Interoperability among ontologies: available solutions

Several solutions have been proposed in literature to address the ontology interoperability problem. Different methods tackle different aspects of the problem.

Some of the existing approaches are aimed at enabling interoperability at the language level. It means that they try to map the formalisms used to represent the ontologies, in order to homogenize the descriptions and then to compare the elements belonging to the different ontologies (OntoMorph provides transformation rules between languages constructs, using OKBC ontologies can be mapped to a common knowledge model).

Several approaches are interactive and suggest to the user possible alignments and mapping. Among them, we can find linguistic based approaches and approaches based on the analysis and evaluation of the structural and model similarity.

There are also tools that provides additional services, such as results diagnosis and checking. For instance Chimaera allows to perform domain independent validation checks, based on heuristics; in OntoMorph and in PROMPT reasoning based verification is implemented.

Finally, some tools (e.g., SHOE) have some features aimed at supporting ontology evolution; interesting services are the possibility of revising and integrating definitions without invalidating existing ontologies, the possibility of creating a new ontology that extends existing ones or that is the result of the intersection of existing ones, the possibility of performing mutual revisions of ontologies.

In the following, a detailed description of the most relevant solution for interoperability among ontologies available in literature is presented. Some features are put in evidence in a table.

1.4.1 FCA- Merge

Implemented by	AIFB - Karlsruhe University		
Type of solution	Tool [ ]	Method [x]	
	Tool name:		
	Version:		
Date of issue/developm.	2001		
Last update			
Level	Concepts [ ]	Instances [x]	
Problem addressed	Merging [x]	Mapping [ ]	Alignement [ ]
Level of automation	Manual [ ]	Semi-automated [ ]	Fully Automated [x]
Language(s) supported			
Description	Fca-Merge [SM01] is a method proposed by Stumme and Maedche for ontology merging. For the source ontologies, it extracts instances from a given set of domain-specific text documents by applying natural language processing techniques. Based on the extracted instances, mathematically founded techniques are applied taken from *Formal Concept Analysis* [GW99] to derive a lattice of concepts as a structural result. The produced result is explored and transformed to the merged ontology by the ontology engineer. They use lexical analysis to perform, among other things, retrieval of domain-specific information.		
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	Techniques from natural language processing and formal concept analysis are applied		

1.4.2 IF – map

Implemented by			
Type of solution	Tool [ ]	Method [x]	
	Tool name:		
	Version:		
Date of issue/developm.	2002		
Last update			
Level	Concepts [x]	Instances [x]	
Problem addressed	Merging [ ]	Mapping [x]	Alignement [ ]
Level of automation	Manual [ ]	Semi-automated [ ]	Fully Automated [x]

Language(s) supported	RDF, KIF, Ontolingua, to native Protege KBs and Prolog KB
Description	IF - map is an automatic method for ontology mapping developed by Kalfoglou and Schorlemmer [KS02] based on the Barwise-Seligman theory of information flow [BS99]. Their method draws on the proven theoretical ground of Barwise and Seligman's channel theory, and provides a systematic and mechanised way for deploying it on a distributed environment to perform ontology mapping among a variety of different ontologies. These mappings are formalised in terms of *logic infomorphisms* An infomorphism is a morphism between *local logics*; every ontology have associated a *local logic*, a triple (set of instances, set of types, classification relation) . IF-Map is declaratively specified in Horn logic and executed with a standard Prolog engine. Therefore, we partially translate a variety of input format to Horn clauses with the aim of a customised translator. We deal with a variety of formats ranging from RDF, KIF, Ontolingua, to native Protege KBs and Prolog KB. Infomorphisms are expressed in RDF.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	

1.4.3 PROMPT

Implemented by	Stanford University		
Type of solution	Tool ☑	Method ☐	
	Tool name: PROMPT Suite		
	Version:2.1.3		
Date of issue/developm.	2000		
Last update	2004		
Level	Concepts ☑	Instances ☐	
Problem addressed	Merging ☑	Mapping ☑	Alignement ☐
Level of automation	Manual ☐	Semi-automated ☑	Fully Automated ☐
Language(s) supported	OKBC		
Description	**PROMPT** [NM00] is a tool for the *Protegè* ontology development environment developed from Noy and Musen. The knowledge model underlying PROMPT is frame-based and it is compatible with OKBC [CF98]. At the top level, there are classes, slots, facets, and instances: · **Classes** are collections of objects that have similar properties. Classes are arranged into a subclass–superclass hierarchy with multiple inheritance. Each class has slots attached to it. Slots are inherited by the subclasses. · **Slots** are named binary relations between a class and either another class or a primitive object (such as a string or a number). Slots attached to a class may be further constrained by facets.		

	· **Facets** are named ternary relations between a class, a slot, and either another class or a primitive object. Facets may impose additional constraints on a slot attached to a class, such as the cardinality or value type of a slot. · **Instances** are individual members of classes. These definitions are the only restrictions that they impose on the input ontologies for PROMPT. Since this knowledge model is extremely general, and many existing knowledge representation systems have knowledge models compatible with it, the solutions to merging and alignment produced by PROMPT can be applied over a variety of knowledge representation systems. PROMPT takes two ontologies as input and guides the user in the creation of one merged ontology as output. First PROMPT creates an initial list of matches based on class names. Then the following cycle happens: (1) the user triggers an operation by either selecting one of PROMPT's suggestions from the list or by using an ontology-editing environment to specify the desired operation directly; and (2) PROMPT performs the operation, *automatically* executes additional changes based on the type of the operation, generates a list of *suggestions* for the user based on the structure of the ontology around the arguments to the last operation, and determines *conflicts* that the last operation introduced in the ontology and finds possible solutions for those conflicts.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	iterative suggestions for concept merges based on linguistic and structural knowledge

1.4.4 Prompt-Diff

Implemented by	Stanford University		
Type of solution	Tool ☑	Method ☐	
	Tool name: Prompt Suite		
	Version: 2.1.3		
Date of issue/developm.	2002		
Last update	2004		
Level	Concepts ☑	Instances ☐	
Problem addressed	Merging ☑	Mapping ☑	Alignement ☐
Level of automation	Manual ☐	Semi-automated ☐	Fully Automated ☐
Language(s) supported	OKBC		
Description	The PromptDiff algorithm combines an arbitrary number of heuristic matchers, each of which looks for a particular property in the unmatched frames. All the matchers must conform to the monotonicity principle: Matchers do not retract any matches already in the table. They may delete the rows where one of the frames is null by creating new matches. But it must keep all the existing matches.		

Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	Designed for Ontology versioning

1.4.5 CHIMAERA

Implemented by	KSL, Stanford University		
Type of solution	Tool ☑	Method ☐	
	Tool name:		
	Version:		
Date of issue/developm.	2000		
Last update			
Level	Concepts ☑	Instances ☐	
Problem addressed	Merging ☑	Mapping ☐	Alignement ☐
Level of automation	Manual ☐	Semi-automated ☑	Fully Automated ☐
Language(s) supported			
Description	Chimaera [MF00] is an interactive tool by McGuinness et al., and the engineer is in charge of making decisions that will affect the merging process. Chimaera analyses the ontologies to be merged, and if linguistic matches are found, the merge is done automatically, otherwise the user is prompted for further action. It appears to be similar to PROMPT; they are both embedded in ontology editing environments and give interactive suggestions to the user.		
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	It solves mismatches at the terminological level in a very light way, provides interactive suggestions to the users, provides diagnostic function.		

1.4.6 GLUE

Implemented by	Univ. Of Washington		
Type of solution	Tool ☑	Method ☑	
	Tool name: GLUE		
	Version:		
Date of issue/developm.	2002		
Last update	2004		
Level	Concepts ☑	Instances ☑	
Problem addressed	Merging ☐	Mapping ☑	Alignement ☐

Level of automation	Manual ☐ Semi-automated ☐ Fully Automated ☑
Language(s) supported	
Description	Doan *et al.* [DM02] developed a system, **GLUE**, which employs machine learning techniques to find mappings. Given two ontologies, for each concept in one ontology, GLUE finds the most similar concept in the other ontology using probabilistic definitions of several practical similarity measures. The authors claim that this is their difference when comparing their work with other machine-learning approaches, where only a single similarity measure is used. In addition to this, GLUE also uses multiple learning strategies, each of which exploits a different type of information either in the data instances or in the taxonomic structure of the ontologies . The similarity measure they employ is the joint probability distribution of the concepts involved, so instead of committing to a particular definition of similarity, GLUE calculates the joint distribution of the concepts, and lets the application use the joint distribution to compute any suitable similarity measure. The authors says that GLUE uses a multi-learning strategy; a learner can inspect instances in many different ways. It can exploit the frequencies of words in the text value of instances, the instance names, the value formats, or the characteristics of value distributions. The authors developed two learners, a **content learner** and a **name learner**. The content learner uses a text classification method, called Naive Bayes larning. The name learner is similar to the content learner but uses the full name of the instance instead of its content. Then there's another learner called the **meta-learner** that combines the predictions of the two learners assigning to each one of them a learner weight that indicates how much its predictions is trustable. The authors also used a technique, relaxation labelling, that assigns labels to nodes of a graph, given a set of constraints. The authors applied this technique to map two ontologies taxonomies, *O*1 to *O*2, by regarding concepts (nodes) in *O*2 as labels, and recasting the problem as finding the best label assignment to concepts (nodes) in *O*1, given all knowledge they have about the domain and the two taxonomies. That knowledge can include domain-independent constraints like "two nodes match if nodes in their neighbourhood also match" – where neighbourhood is defined to be the children, the parents or both – as well as domain-dependent constraints like "if node Y is a descendant of node X, and Y matches professor, then it is unlikely that matches assistant professor". The system has been empirically evaluated with mapping two university course catalogues.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	It uses machine learning strategies and probabilistic measures of similarity.

1.4.7 CAIMAN

Implemented by	
Type of solution	Tool ☑ Method ☐

	Tool name: Version:
Date of issue/developm.	2001
Last update	
Level	Concepts ☑ / Instances ☑
Problem addressed	Merging ☐ / Mapping ☑ / Alignement ☐
Level of automation	Manual ☐ / Semi-automated ☐ / Fully Automated ☐
Language(s) supported	
Description	The ontology mapping in CAIMAN [LG01] is based on a approach which considers the concepts in an ontology implicitly represented by the documents assigned to each concept. Using machine learning techniques for text classification, a concept in a personal ontology is mapped to a concept in community ontology.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	Use machine learning methods

1.4.8 ONION

Implemented by	Stanford University
Type of solution	Tool ☑ / Method ☑ Tool name: OnTo-Agents toolkit Version: under development
Date of issue/developm.	2000
Last update	
Level	Concepts ☑ / Instances ☐
Problem addressed	Merging ☑ / Mapping ☐ / Alignement ☐
Level of automation	Manual ☐ / Semi-automated ☐ / Fully Automated ☑
Language(s) supported	
Description	The ONION (ONtology compositiON) [MW00] system was developed by Mitra, Wiederhold et al. in the Stanford University Database Group. They propose a scalable and mantainable approach based on the use rules that creates an *articulation* or linkage between the systems. The rules are generated using a semi-automatic articulation tool with the help of a domain expert. To make the sources ontologies compliant for automatic composition, based on the accumulated knowledge rules, they represent them using a graph-oriented model extended with a small algebraic operator set. For example, one information source may use UML as modelling language and another using DAML+OIL, ONION will convert the ontologies

	associated with both information sources to the ONION *conceptual model*. They intend to support a small number of classes of such ontology models that are in use providing wrappers which will convert from these models to the ONION format. Conversion of ontologies from their native models to the ONION format can be also be performed declaratively generating rules that correlate parts of one ontology to parts of another based on semantic similarity.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	

1.4.9 Breis and Bejar Framework

Implemented by	Fernández-Breis and Martínez-Béjar		
Type of solution	Tool ☐	Method ☑	
	Tool name:		
	Version:		
Date of issue/developm.	2002		
Last update			
Level	Concepts ☑	Instances ☐	
Problem addressed	Merging ☑	Mapping ☐	Alignement ☐
Level of automation	Manual ☐	Semi-automated ☐	Fully Automated ☑
Language(s) supported			
Description	**Fernández-Breis and Martínez-Béjar** [FM02] describe a cooperative framework for integrating ontologies. Their system is aimed towards ontology integration and is intended for use by normal and expert users. The former are seeking information and provide specific information with regard to their concepts, whereas the latter are integration-derived ontology constructors, in the authors' jargon. As the normal users enter information regarding the concepts' attributes, taxonomic relations and associated terms in the system, the expert users process this information and the system helps them to derive the integrated ontology. The algorithm that supports this integration is based on taxonomic features and on detection of synonymous concepts in the two ontologies. It also takes into account the attributes of concepts and the authors have defined a typology of equality criteria for concepts. For example, when the name-based equality criterion is called upon, both concepts must have the same attributes.		
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)			

1.4.10 MAFRA

Implemented by	Karlsruhe University		
Type of solution	Tool ☑	Method ☐	
	Tool name: MApping FRAmework		
	Version: 0.2		
Date of issue/developm.			
Last update	2003		
Level	Concepts ☑	Instances ☑	
Problem addressed	Merging ☐	Mapping ☑	Alignement ☐
Level of automation	Manual ☐	Semi-automated ☑	Fully Automated ☐
Language(s) supported			
Description	**MAFRA** [MM02]is part of a multi-ontology system, and it aims to automatically detect similarities of entities contained in two different department ontologies. Both ontologies must be normalized to a uniform representation, eliminating syntax differences and making semantic differences between the source and the target ontology more apparent. This normalisation process is done by a tool, LIFT, which brings DTDs, XML-Schema and relational databases to the structural level of the ontology. Another interesting contribution of the MAFRA framework is the definition of a *semantic bridge*. This is a module that establishes correspondences between entities from the source and target ontology based on similarities found between them. All the information regarding the mapping process is accumulated, and populate an ontology of mapping constructs, the so called *Semantic Bridge Ontology* (SBO). The SBO is in DAML+OIL format, and the authors argue, one of the goals in specifying the semantic bridge ontology was to maintain and exploit the existent constructs and minimize extra constructs, which could maximize as much as possible the acceptance and understanding by general semantic web tools.		
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	The tool tries to identify similarity between entities belonging to the source and the target ontology and allows to establish a correspondence between these entities. Transformation rules for translating the instances can be derived from the established mappings and executed. Cooperative mapping facilities are provided.		

1.4.11 OIS

Implemented by	Calvanese, De Giacomo, Lenzerini	
Type of solution	Tool ☐	Method ☑
	Tool name:	
	Version:	
Date of issue/developm.	2001	

Last update			
Level	Concepts ☑	Instances ☐	
Problem addressed	Merging ☑	Mapping ☑	Alignement ☐
Level of automation	Manual ☐	Semi-automated ☐	Fully Automated ☑
Language(s) supported	DL		
Description	Calvanese *et al.* [CD01] proposed a formal framework for Ontology Integration Systems – **OISs**. The framework provides the basis for ontology integration, which is the main focus of their work. Their view of a formal framework is deals with a situation where we have various local ontologies, developed independently from each other, assisting the task to build an integrated, global ontology as a means for extracting information from the local ones. Ontologies in their framework are expressed as Description Logic (DL) knowledge bases, and mappings between ontologies are expressed through suitable mechanisms based on queries. Although the framework does not make explicit any of the mechanisms proposed, they are employing the notion of queries, which allow for mapping a concept in one ontology into a view, i.e., a query, over the other ontologies, which acquires the relevant information by navigating and aggregating several concepts. They propose two approaches to realise this query/view-based mapping: global-centric and local-centric. The global-centric approach is an adaptation of most data integration systems. In such systems, the authors continue, sources are databases, the global ontology is actually a database schema, and the mapping is specified by associating to each relation in the global schema one relational query over the source relations. In contrast, the local-centric approach requires reformulation of the query in terms of the queries to the local sources.		
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)			

1.4.12 Madhavan et al. Framework

Implemented by			
Type of solution	Tool ☑	Method ☑	
	Tool name: CUPID		
	Version:		
Date of issue/developm.	2001		
Last update			
Level	Concepts ☑	Instances ☐	
Problem addressed	Merging ☐	Mapping ☑	Alignement ☐

Level of automation	Manual ☐ / Semi-automated ☐ / Fully Automated ☐
Language(s) supported	
Description	Madhavan *et al.* [MB01] developed a framework and propose a language for ontology mapping. Their framework enables mapping between models in different representation languages without first translating the models into a common language, the authors claim. The framework uses a *helper model* when it is not possible to map directly between a pair of models, and it also enables representing mappings that are either incomplete or involve loose information. The models represented in their framework are representations of a domain in a formal language, and the mapping between models consists of a set of relationships between expressions over the given models. The expression language used in a mapping varies depending on the languages of the models being mapped. The authors claim that mapping formulae in their language can be fairly expressive, which makes it possible to represent complex relationships between models. They applied their framework in an example case with relational database models. They also define a typology of mapping properties: query answerability, mapping inference and mapping composition. The authors argue that a mapping between two models rarely maps all the concepts in one model to all concepts in the other. Instead, mappings typically lose some information and can be partial or incomplete. Question answerability is a proposed formalisation of this property. Mapping inference provides a tool for determining types of mapping, namely equivalent mappings and minimal mappings; and mapping composition enables one to map between models that are related by intermediate models.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	

1.4.13 Kiryakov *et al.* Framework

Implemented by	
Type of solution	Tool ☐ / Method ☑ Tool name: Version:
Date of issue/developm.	2001
Last update	
Level	Concepts ☑ / Instances ☐
Problem addressed	Merging ☐ / Mapping ☑ / Alignement ☐
Level of automation	Manual ☐ / Semi-automated ☐ / Fully Automated ☐
Language(s) supported	

Description	Kiryakov *et al.* [KD01] developed a framework for accessing and integrating upper-level ontologies. They provide a service that allows a user to import linguistic ontologies onto a Web server, which will then be mapped onto other ontologies. The authors argue for a uniform representation of the ontologies and the mappings between them, a relatively simple meta-ontology (*OntoMapO*) of property types and relation-types should be defined. Apart from the OntoMapO primitives and design style, the authors elaborate on a set of primitives that OntoMapO offers for mapping. There are two sets of primitives defined, *InterOntologyRel* and *IntraOntologyRel*, each of which has a number of relations that aim to capture the correspondence of concepts originating from different ontologies (i.e. equivalent, more specific, meta-concept). A typology of these relations is given in the form of a hierarchy and the authors claim that an initial prototype has been used to map parts of the *CyC* ontology to *EuroWordNet*.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	

1.4.14 IFF

Implemented by	Robert E. Kent		
Type of solution	Tool ☑	Method ☑	
	Tool name:IFF		
	Version:		
Date of issue/developm.			
Last update			
Level	Concepts ☑	Instances ☑	
Problem addressed	Merging ☐	Mapping ☑	Alignement ☐
Level of automation	Manual ☐	Semi-automated ☐	Fully Automated ☐
Language(s) supported			
Description	Kent (2000) proposed a framework for ontological structures to support ontology sharing. It is based on the Barwise-Seligman theory of information flow [GW99]. Kent argues that **IFF** represents the dynamism and stability of knowledge. The former refers to instance collections, their classification relations and links between ontologies specified by ontological extension and synonymy (type equivalence); it is formalised with Barwise and Seligman's *local logics* and their structure-preserving transformations – logic infomorphisms. Stability refers to concept/relation symbols and to constraints specified within ontologies; it is formalised with Barwise and Seligman's *regular theories* and their structure-preserving transformations. IFF represents ontologies as logics, and ontology sharing as a specifiable ontology extension hierarchy. An ontology, Kent continues, has a classification relation between instances and concept/relation symbols, and		

	also has a set of constraints modelling the ontology's semantics. In Kent's proposed framework, a community ontology is the basic unit of ontology sharing; community ontologies share terminology and constraints through a common generic ontology that each extends, and these constraints are consensual agreements within those communities. Constraints in generic ontologies are also consensual agreements but across communities.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	Reasoning on type equivalence. Specific for ontology sharing

1.4.15 ODEMerge

Implemented by	Universidad Politécnica de Madrid		
Type of solution	Tool ☑	Method ☑	
	Tool name: WEBOde		
	Version: 2.0.2		
Date of issue/developm.			
Last update	2003		
Level	Concepts ☑	Instances ☐	
Problem addressed	Merging ☑	Mapping ☐	Alignement ☐
Level of automation	Manual ☐	Semi-automated ☐	Fully Automated ☐
Language(s) supported	Importing ontologies written in XML, RDF(S) or CARIN, and exporting into XML, RDF(S), OIL, DAML+OIL, CARIN, FLogic, Prolog, Jess, Java and HTML		
Description	ODEMerge [CP01] is a tool to merge ontologies that is integrated in **WebODE**, the software platform to build ontologies that has been developed by the Ontology Group at Technical University of Madrid. It is a client-server tool that works in the Web. This tool is a partial software support for the **methodology** for merging ontologies elaborated by de Diego [D01]. This methodology proposes the following steps: 1) transformation of formats of the ontologies to be merged; 2) evaluation of the ontologies; 3) merging of the ontologies; 4) evaluation of the result; 5) transformation of the format of the resulting ontology to be adapted to the application where it will be used. The methodology exposes in a very detailed way the sequence of steps that must be executed to perform the task of merging two ontologies, who have to perform each step, how he has to perform it, and which should be the products of such steps. For the evaluation and merging of ontologies, very detailed rules are proposed. The methodology is based on the experience merging e-commerce ontologies.		

	WebODE helps in steps (1), (2), (4) and (5) of the merging methodology, and ODEMerge carries out the merge of taxonomies of concepts in step (3). Besides, ODEMerge helps in the merging of attributes and relations, and it incorporates many of the rules identified in the methodology. ODEMerge uses the following inputs: • the source ontology 1 to be merged; • the source ontology 2 to be merged; • the *table of synonyms*, which contains the synonymy relationships of the terms of ontology 1 with the terms of the ontology 2. • the *table of hyperonyms*, which contains the hyperonymy relationships of the terms of ontology 1 with the terms of the ontology 2. ODEMerge processes the ontologies together with the information of the tables of synonymy and hyperonymy, and it generates a new ontology, which is the merge of the ontology 1 and the ontology 2. New versions of the tool will include electronic dictionaries and other linguistic resources that can substitute the tables of synonyms and hyperonyms. This tool can be easily **extensible** to consider new rules of merging that can be identified. Another important characteristic of ODEMerge is that it can be used to merge ontologies in a large number of **ontology implementation languages**. The WebODE import module allows importing ontologies written in XML, RDF(S) or CARIN, and allows exporting into XML, RDF(S), OIL, DAML+OIL, CARIN, FLogic, Prolog, Jess, Java and HTML.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	Use of a common model of representation of the ontologies.

1.4.16 HELIOS

Developed by	Università degli Studi di Milano		
Type of solution	Tool ☑	Method ☑	
	Tool name: HELIOS		
	Version:		
Date of issue/developm.			
Last update			
Level	Concepts ☑	Instances ☐	
Problem addressed	Merging ☑	Mapping ☑	Alignement ☐
Level of automation	Manual ☐	Semi-automated ☐	Fully Automated ☑
Language(s) supported	RDF(S), DAML+OIL, OWL, ODLi3		
Description	HELIOS (Helios Evolving Interaction-based Ontology knowledge Sharing)		

	[CF03b] is a framework for supporting dynamic ontology-based knowledge sharing and evolution in P2P networks and more generally in open distributed systems. The knowledge sharing and evolution processes in HELIOS are based on peer ontologies, describing the knowledge of each peer (that is, the knowledge a peer brings to the network and the knowledge the peer has of network), and on interactions among peers, allowing information search and knowledge acquisition/extension, according to pre-defined query models and dynamic ontology matching techniques. The matching techniques implemented in HELIOS are based on the H-MATCH [CF03] algorithm which considers both the linguistic features and contextual features of concepts in the ontology of a given node. Linguistic features are constituted by the semantic content of terms used as names of concepts and properties. Contextual features are constituted by the concept properties and adjacent concepts (i.e., concepts having a semantic relation with the considered concept). The ontology matching process is based on a thesaurus, where the meaning of each term in the ontology of a given node is represented by the set of terminological relationships that it has with other terms. The thesaurus is built by exploiting WordNet as a reference source of lexical information. H-MATCH provides four different matching models that are used for dynamically suiting the matching process to different levels of richness in ontology descriptions. H-MATCH is used in HELIOS in order to enable knowledge sharing and resource discovery in open distributed systems. When a peer receives a query from another peer, the query is processed against its own ontology in order to extract the target concept(s) and the matching model to use. Once concepts matching a target concept have been identified using H-MATCH, they are returned to the requesting peer through a query answer. A detailed description of HELIOS and H-MATCH is provided in [CF03, CF03b]
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	Dynamic ontology matching algorithm based on linguistic and contextual features of concepts for evaluating concept similarity; absence of an a priori agreement among peers for ontology specification; use of WordNet for linguistic matching.

1.4.17 ARTEMIS

Developed by	Università degli Studi di Milano – Università degli Studi di Brescia	
Type of solution	Tool ☑	Method ☑
	Tool name: ARTEMIS	
	Version:	
Date of issue/developm.		
Last update		
Level	Concepts ☑	Instances ☐

Problem addressed	Merging ☐	Mapping ☑	Alignement ☐
Level of automation	Manual ☐	Semi-automated ☐	Fully Automated ☑
Language(s) supported	ODLi3		
Description	The Artemis tool environment performs the semantic integration of strongly heterogeneous data sources both structured and semi-structured The integration process is based on the construction of a semantically rich representation of the data sources to be integrated by means of a common data model based on the ODLi3 language and allows the construction of a global integrated view of data at different sources. ARTEMIS exploits interschema knowledge expressed through intensional properties and extensional properties. All interschema properties are stored in a thesaurus. Three different alternatives are supported in ARTEMIS for building the thesaurus: i) to use a domain-dependent thesaurus manually constructed by the designer; ii) to use a domain-independent thesaurus extracted from the WordNet lexical system; iii) to use a hybrid thesaurus, which is a combination of the previous two. The integration process is performed by the ARTEMIS mediator environment, which is composed by schema matching and unification environments. Schema matching has the goal of identifying schema elements candidate to integration, that is, schema elements that describe the same or semantically related information in different source schemas. Schema matching in ARTEMIS is performed through affinity-based and clustering techniques based on the interschema knowledge stored in the thesaurus. The output of clustering is an affinity tree, where classes are the leaves and intermediate nodes have an associated affinity value, holding for all cluster members. Clusters to be unified (i.e.,candidate clusters) are interactively selected from the affinity tree using a threshold-based mechanism. Schema unification has the goal of defining the mediation scheme as a collection of global views, out of candidate clusters in the affinity tree. The unification process is performed through rule-based techniques, by which names and properties of schema elements in a cluster are properly reconciled into a global, unified representation. To support global query rewriting, a set of mapping rules are also specified for each global view, stating how to map each global property onto local properties of schema elements of the provenance cluster. The ARTEMIS integration process and associated tool environment are pioneer in proposing affinity-based metrics and clustering procedures for schema matching and integration. Such innovative and featuring techniques have been subsequently imported and further refined in developing the MOMIS system, whose schema matching process is entirely based on the ARTEMIS techniques. Moreover, ARTEMIS has been recently extended to support domain ontology construction by extracting ontological concepts and semantic relationships among them from global views. Ontological concepts are defined according to different structures of global views. Three types of semantic relationships are considered: generalization (is-a), disjunction (two disjoint concepts have disjoint sets of their instances), equivalence (equivalent concepts have the same sets of instances). Concepts and semantic relationships are exploited to implement semantic search			

	modalities in a given domain. In particular, in [BD04] is presented a related ontology-based approach to support effective use and sharing of knowledge coming from several organizations to enhance communication intra and inter organizations. For a theoretical overview of the foundation of ARTEMIS refer to [AC01].
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	Affinity based clustering techniques for performing schema matching; rule-based unification techniques for global views construction; use of interschema properties and terminological relationships.

1.4.18 SWAP

Developed by	University of Karlsruhe		
Type of solution	Tool ☑	Method ☑	
	Tool name: SWAP		
	Version:		
Date of issue/developm.			
Last update			
Level	Concepts ☑	Instances ☐	
Problem addressed	Merging ☑	Mapping	Alignement ☐
Level of automation	Manual ☐	Semi-automated ☐	Fully Automated ☑
Language(s) supported	RDF, RDF(S), SeRQL		
Description	In the project SWAP (Semantic Web and Peer-to-Peer) [ET03] semantic descriptions of datasources stored by peers and semantic descriptions of peers themselves are exploited for formulating queries such that they can be understood by other peers, for merging the answers received from other peers, and for routing queries across the network. In particular, ontologies and Semantic Web techniques have been used for the semantic descriptions of contents and queries in the P2P system. To this purpose an RDF(S) metadata model for encoding semantic information is introduced allowing peers to handle heterogeneous and even contradictory views on the domain of interest. Each peer implements an ontology extraction method to extract from its different information sources an RDF(S) description (ontology) compatible with the SWAP metadata model. Such ontologies are used by the SeRQL Query Language to perform query processing; peers storing knowledge semantically related to a target concept are localized through SeRQL views defined on specific similarity measures. Views from external peers are integrated through an ontology merging method to extend the knowledge of the receiving peer according to a specific rating model.		
Special Features (e.g., learning method; use of a reference ontology,	Use of a RDF(S) model for ontology specification; use of the SeRQL query language for query resolution over ontologies.		

similarity reasoning, rule-base editable)	

1.4.19 MOMIS

Implemented by	University of Modena, University of Milano, University of Brescia
Type of solution	Tool ☑ / Method ☑ Tool name: ODB tools+ ARTEMIS Version:
Date of issue/developm.	1999
Last update	
Level	Concepts ☑ / Instances ☐
Problem addressed	Merging ☑ / Mapping ☐ / Alignement ☐
Level of automation	Manual ☐ / Semi-automated ☑ / Fully Automated ☐
Language(s) supported	
Description	The MOMIS (Mediator envirOnment for Multiple Information Sources) [BC99] is a framework to perform information extraction and integration from both structured and semistructured data sources. An object-oriented language, with an underlying Description Logic, called ODL-I3, derived from the standard ODMG is introduced for information extraction. Information integration is then performed in a semi-automatic way, by exploiting the knowledge in a Common Thesaurus (defined by the framework) and ODL-I3 descriptions of source schemas with a combination of clustering techniques and Description Logics. This integration process gives rise to a virtual integrated view of the underlying sources (the Global Schema) for which mapping rules and integrity constraints are specified to handle heterogeneity. The MOMIS system, based on a conventional wrapper/mediator architecture, provides methods and open tools for data management in Internet-based information systems by using a CORBA-2 interface.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	

1.4.20 SHOE

Implemented by	University of Maryland
Type of solution	Tool ☑ / Method ☐ Tool name: Knowledge Annotator, Expose' (http://www.cs.umd.edu/projects/plus/SHOE/) Version:
Date of issue/developm.	1997

Last update	no longer being actively maintained
Level	Concepts [x] / Instances [ ]
Problem addressed	Merging [x] / Mapping [ ] / Alignement [ ]
Level of automation	Manual [ ] / Semi-automated [ ] / Fully Automated [ ]
Language(s) supported	SHOE
Description	SHOE [LS97] is a superset of HTML which adds the tags necessary to embed arbitrary semantic data into web pages. SHOE tags are divided into two categories. First, there are tags for constructing ontologies. SHOE ontologies are sets of rules which define what kinds of assertions SHOE documents can make and what these assertions mean. For example, a SHOE ontology might say that a SHOE document can declare that some data entity is a 'dog', and that if it does so, that this 'dog' is permitted to have a 'name'. Secondly, there are tags for annotating web documents to subscribe to one or more ontologies, declare data entities, and make assertions about those entities under the rules proscribed by the ontologies. The mergin fo different ontologies is obtained by using inference rules, defined to map the common items between the ontologies to be merged. Terminological differences are solved by defining if-and-only-if rules; Scope differences are solved by specifying mapping to most specific category (based on subsumption); encoding differences are handled by mapping individual values.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	SHOE allows to prove some results of ontology difference, but ontologies must be written in the same language (SHOE)

1.4.21 INFOSLEUTH

Implemented by	Telcordia technologies
Type of solution	Tool [x] / Method [ ] Tool name: Version:
Date of issue/developm.	
Last update	
Level	Concepts [ ] / Instances [ ]
Problem addressed	Merging [x] / Mapping [x] / Alignement [ ]
Level of automation	Manual [ ] / Semi-automated [ ] / Fully Automated [ ]
Language(s) supported	OKBC
Description	InfoSleuth [NF99] is an agent-based system that can be configured to perform many different information management activities in a distributed

	environment. InfoSleuth agents provide a number of complex query services that require resolving ontology-based queries over dynamically changing, distributed, heterogeneous resources. These include distributed query processing, location-independent single-resource updates, event and information monitoring, statistical or inferential data analysis, and trend discovery in complex event streams. It has been used in numerous applications, including the Environmental Data Exchange Network and the Competitive Intelligence System. Ontologies are specified in OKBC [CF98] and are stored in an OKBC server and accessed via ontology agents. These agents provide ontology specifications to users for request formulation, to resource agents for mapping and to other agents that need to understand and process requests and information in the application domain.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	Agent based

1.4.22 KRAFT

Developed by	Alun Preece, Trevor Bench-Capon, Dean Jones et al. http://www.csd.abdn.ac.uk/research/kraft.html		
Type of solution	Tool ☑	Method ☑	
	Tool name: KRAFT		
	Version:		
Date of issue/developm.	1997		
Last update			
Level	Concepts ☐	Instances ☐	
Problem addressed	Merging ☑	Mapping ☑	Alignement ☐
Level of automation	Manual ☐	Semi-automated ☐	Fully Automated ☐
Language(s) supported			
Description	In KRAFT [PH99] translations between different ontologies are done by special mediator agents which can be customized to translate between different ontologies and 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 aims at reaching 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.		
Special Features (e.g., learning method; use of a reference ontology,	Use of customizable mediator agents		

similarity reasoning, rule-base editable)	

1.4.23 ONIONS

Developed by	CNR, Conceptual Modeling Group, Rome
Type of solution	Tool ☐ / Method ☑ Tool name: Version:
Date of issue/developm.	1996
Last update	
Level	Concepts ☐ / Instances ☐
Problem addressed	Merging ☑ / Mapping ☐ / Alignement ☐
Level of automation	Manual ☐ / Semi-automated ☐ / Fully Automated ☐
Language(s) supported	
Description	ONIONS [GS96] methodology can be summarized in the following 6 phases: • 1: Creating a corpus of validated textual sources of a domain. Sources must be individuated together with an assessment of their diffusion and validation inside the domain community. • 2: Taxonomic analysis. If lacking, taxonomies are constructed. • 3: Local source analysis. The conceptual analysis of terms in order to locate their free-text descriptions and other constraints (local definitions). • 4: Multi-local source analysis. The conceptual analysis of the descriptions allows to link the local definitions with multi-local concepts and general knowledge (paradigms). • 5: Building an integrated ontology library. An ontology library covers all the local definitions and the paradigms that have been used in building multi-local, integrated definitions. • 6: Implementing and classifying the library. These steps pertain to the diffusion, use, classification, and validation of the model.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	

1.4.24 OBSERVER

Developed by	E. Mena V. Kashyap A. Sheth A. Illarramendi
Type of solution	Tool ☑ / Method ☐ Tool name: OBSERVER Version:
Date of issue/developm.	1996
Last update	2003
Level	Concepts ☑ / Instances ☐
Problem addressed	Merging ☐ / Mapping ☑ / Alignement ☐
Level of automation	Manual ☑ / Semi-automated ☐ / Fully Automated ☐
Language(s) supported	
Description	OBSERVER [MK96] is a system developed by E. Mena, V. Kashyap, A. Sheth and A. Illarramendi. In order to access heterogeneous data repositories, the objects in such repositories are represented as intensional descriptions by pre-existing ontologies expressed in Description Logics characterizing information in different domains. User queries are rewritten by using interontology relationships to obtain semantic preserving translations across the ontologies. There are two types of mappings: one type links each term in an ontology with structures in data repositories are used in order to access and retrieve data from the repositories; the other type (Interontology Relationships Manager) relates the terms in various ontologies.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	Query rewriting

1.4.25 ONTOMORPH

Developed by	Hans Chalupsky
Type of solution	Tool ☑ / Method ☐ Tool name: LOOM, PowerLoom Version: 4.0
Date of issue/developm.	2000

Last update	2004
Level	Concepts ☑ Instances ☐
Problem addressed	Merging ☑ Mapping ☐ Alignement ☐
Level of automation	Manual ☑ Semi-automated ☐ Fully Automated ☐
Language(s) supported	
Description	OntoMorph [C00] provides a powerful rule language to represent complex syntactic transformations and a rule interpreter to apply them to arbitrary KR language expressions. OntoMorph is fully integrated with the PowerLoom KR system to allow transformations based on any mixture of syntactic and semantic criteria. OntoMorph's successful application as an input translator for a critiquing system and as the core of a translation service for agent communication. We further show how knowledge base merging can be cast as a translation problem and motivate how OntoMorph can be applied to knowledge base merging tasks.
Special Features (e.g., learning method; use of a reference ontology, similarity reasoning, rule-base editable)	Syntactic/Semantic rewriting

1.5 MultiOntology Architectures & Environments, Federated ontologies

One of the most interesting aspects of the Semantic Web is its idea of decentralization. A proposal for software interoperability is that of having a federated system. Such a system expects that every software applications, maintain its data structures modelled by a local ontology, and, in order to support communication and knowledge exchange with others, a mechanism is provided to normalize the local ontologies onto a common ontology model. We report an architecture for federated ontologies as proposed in [SM03].

1. local ontologies (the conceptual models of the autonomous applications), each of them with its specific underlying ontology/metadata repository or database,

2. normalized ontologies (transformation of the local ontologies into a common data model),

3. export ontologies (view on the normalized ontology that describes the relevant parts of the ontology for the federation),

4. one merged ontology (global ontology derived from the combination of the two export schemas), and

5. different applications in the upper layer (external schema layer), which use the merged ontology with their specific views on it.

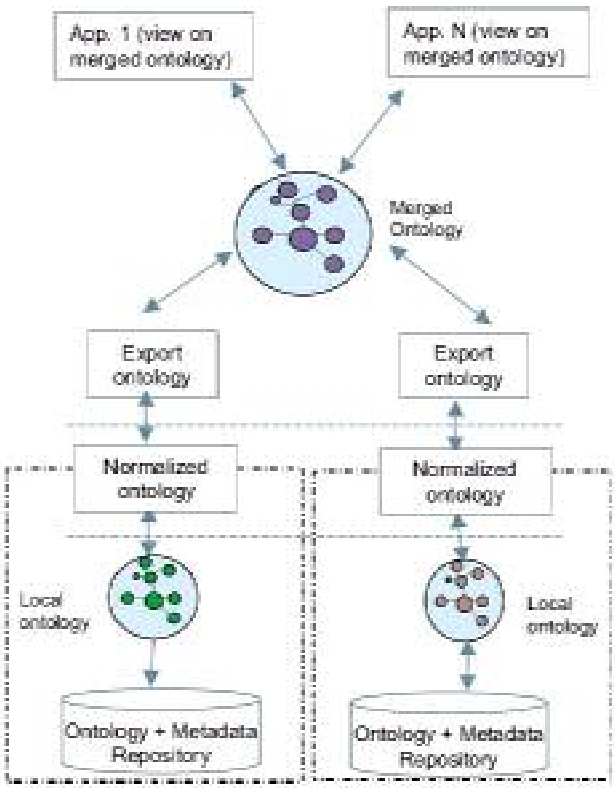

Figure 5: An architecture for federated ontologies

1.6 Contexts and sub-ontology factorization

The notion of context plays a crucial role in different disciplines, such as pragmatics, natural language semantics , linguistics, cognitive psychology, and artificial intelligence (for instance, in NLP techniques, to assign an interpretation to or disambiguate assertions; in distributed AI, contexts are used to design systems of autonomous agents). [Bouquet] In logic, the first representation of context as a formal object was by the philosopher C. S. Peirce; but for nearly eighty years, his treatment was unknown outside its research group. In the 1980s, three different approaches led to related notions of context: Kamp's discourse representation theory; Barwise and Perry's situation semantics; and Sowa's conceptual graphs, which explicitly introduced Peirce's theories to the AI community. During the 1990s, John McCarthy and his students developed a closely related notion of context as a basis for organizing and partitioning knowledge bases. See [SO01] for further details.

For what concerns the application of contexts in the ontology interoperability solutions some solutions consider an ontology as a graph, with a node for each concept and an arc for each relation/attribute; in this perspective they define the "context of a concept" as the neighbourhood of a concept in the ontology graph and use similarity measures based on this kind of contexts.

Other interesting solutions at the state of the art [KS96] propose to evaluate if objects belonging to different ontologies have some semantic similarity with respect to a specific context. This context, called context of comparison, is a reference point that allows to establish the "semantic proximity" of two concepts. this method has been presented for the integration and mapping of database objects, but results very interesting also for ontology mapping purposes.

1.7 Conclusions

We have seen in this chapter, how the field of ontology interoperability is very active; an increasing number of tools and methodologies were developed to address this issue. Unfortunately, at now, no solution seems to be the "winning" one, due to several problems:

- there's a great heterogeneity in the kind of techniques adopted (linguistic similarity, structural similarity, graph matching, heuristics, instance-based similarity,) and it seems that is difficult to find a "balanced mix" of such techniques to have a "general purpose" mapping/merging/alignment tool
- The ontology languages support given by the tools is oftent very limited; to try a tool or a methodology on our own ontology it usually have to be transformed to a common representation format (and we have seen that changing the representation formalism is often not a trivial task)
- Most of the solutions are out-to-date, it is not so easy to understand if a project is still alive, which is the last date of issue/development, not all the tools are still available.

1.8 Bibliography

[AC01] Global Viewing of Heterogeneous Data Sources, 2001; S. Castano (WP8 member), V. De Antonellis (WP8 member), S. De Capitani Di Vimercati

[BC99] Semantic Integration of Semistructured and Structured Data Sources, S. Bergamaschi, S. Castano e M. Vincini, SIGMOD Record Special Issue on Semantic Interoperability in Global Information, Vol. 28, No. 1, March 1999. http://citeseer.ist.psu.edu/bergamaschi99semantic.html

[BD03] The Uni-Level Description: A uniform framework for representing information in multiple data models, 2003; Shawn Bowers; Lois Delcambre

[BD04] Ontology-based interoperability for interorganizational applications, 2004; D. Bianchini; V. De Antonellis; M. Melchiori (authors from UNIBS, INTEROP partner)

[BE03] A Metadata Model for Semantics-Based Peer-to-Peer Systems, 2003; J Broekstra, M Ehrig, P Haase, F. van Harmelen, A. Kampman, M. Sabou, R. Siebes, S. Staab, H. Stuckenschmidt, C. Tempich

[BH01] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. Scienti_c American, 279, 2001

[BM03] A SAT-based Algorithm for Context Matching, 2003; P. Bouquet, B. Magnini, L. Serafini, S. Zanobini

[BS99] Information Flow: the logic of distributed systems, J.Barwise, J.Seligman. Tracts in Theoretical Computer Science 44, Cambridge University Press, ISBN: 0-521-58336-1, 1999

[C00] OntoMorph: A Translation System for Symbolic Knowledge. H. Chalupsky, Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference on Knowledge Representation and Reasoning (KR-2000), Breckenridge, Colorado, USA, April 2000. http://citeseer.ist.psu.edu/chalupsky00ontomorph.html

[CD01] Description logics for information integration. Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. In Computational Logic: From Logic Programming into the Future (In honour of Bob Kowalski), Lecture Notes in Computer Science. SpringerVerlag, 2001. http://citeseer.ist.psu.edu/calvanese01description.html

[CF03] H-MATCH: an Algorithm for Dynamically Matching Ontologies in Peer-based Systems, 2003; S. Castano (WP8 member), A. Ferrara (WP8 member), S. Montanelli (WP8 member)

[CF03b] Helios: A General Framework for Ontology-Based Knowledge Sharing and Evolution in P2P Systems, 2003; S. Castano (WP8 member), A. Ferrara (WP8 member), S. Montanelli (WP8 member), D. Zucchelli

[CF98] OKBC: A Programmatic Foundation for Knowledge Base Interoperability, Vinay K. Chaudhri, Adam Farquhar, Richard Fikes, Peter D. Karp, James P. Rice, Proc. AAAI'98 Conference, Madison, WI, July 1998. http://citeseer.ist.psu.edu/chaudhri98okbc.html

[CO01] Issues in Ontology-based Information Integration, 2001; Zhan Cui Dean Jones Paul O'Brein

[CP01] Solving Integration Problems of E-commerce Standards and Initiatives through Ontological Mappings, Corcho, O. and Gomez-Perez, A. In: Proceedings of the Workshop on E-Business and Intelligent Web at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI2001) , Seattle, USA, August 5, 2001. http://citeseer.ist.psu.edu/corcho01solving.html

[D01] de Diego, R. Método de mezcla de catálogos electrónicos. Final Year Project. Facultad de Informática de la Universidad Politécnica de Madrid. Spain. 2001.

[DM02] Learning to Map between Ontologies on the Semantic Web, A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Proc. of the World-Wide Web Conf. (WWW-2002).

[ET03] Ehrig, M., Tempich, C., Broekstra, J., van Harmelen, F., Sabou, M., Siebes, R., Staab, S., Stuckenschmidt, H.: SWAP - ontology-based knowledge management with peer-to-peer technology. In Sure, Y., Schnurr, H.P., eds.: Proceedings of the 1st National "Workshop Ontologie-basiertes Wissensmanagement (WOW2003)". (2003) To appear 2003. http://citeseer.ist.psu.edu/article/ehrig03swap.html

[FG97] On Ontologies and Enterprise Modelling, 1997; Mark Fox, Michael Gruninger

[FM02] A cooperative framework for integrating ontologies, Jesualdo Tomás Fernández-Breis, Rodrigo Martínez-Béjar,, International Journal of Human-Computer Studies, v.56 n.6, p.665-720, June 2002

[GS96] ONIONS: An Ontological Methodology for Taxonomic Knowledge Integration, GANGEMI, A., STEVE, G. and GIACOMELLI, F. (1996) ECAI-96 Workshop on Ontological Engineering, Budapest, August 13th. http://citeseer.ist.psu.edu/gangemi96onions.html

[GW99] B. Ganter and R. Wille. Formal Concept Analysis: Mathematical Foundations. Springer, Berlin -- Heidelberg, 1999.

[KD01] OntoMap: Ontologies for Lexical Semantics, Atanas Kiryakov, Marin Dimitrov, Kiril Iv. Simov. In: Proceedings of RANLP - 2001"Recent Advances in NLP", 5-7 September 2001, Tzigov Chark, Bulgaria.

[KL01] Michel Klein. Combining and relating ontologies: an analysis of problems and solutions. IJCAI2001 Workshop on Ontologies and Information Sharing, 2001.

[KS02] Information Flow based ontology mapping, Y.Kalfoglou, M.Schorlemmer. In Proceedings of the 1st International Conference on Ontologies, Databases and Application of Semantics (ODBASE'02), Irvine, CA, USA, October 2002

[KS03] Ontology mapping: the state of the art, 2003; Y. Kalfoglou, M. Schorlemmer

[KS96] Semantic and schematic similarities between database objects:a context-based approach, 1996; Vipul Kashyap, Amit Sheth

[KY02] Ontology management - Storing, aligning and maintaining ontologies, 2002; Michel Klein, Ying Ding, Dieter Fensel, Borys Omelayenko

[L01] Ontology and interoperability, 2001; R.Laurini

[LG01] Facilitating the exchange of explicit knowledge through ontology mappings. M. Lacher and G. Groh. In Proceedings of the 1,ith International FLAIRS conference, Key West, FL, USA, May 2001. http://citeseer.ist.psu.edu/lacher01facilitating.html

[LS97] Ontology-based Web agents. S. Luke, L. Spector, D. Rager, and J. Hendler. In W. L. Johnson, editor, Proceedings of the 1st International Conference on Autonomous Agents, pages 59--66. ACM, 1997. http://citeseer.ist.psu.edu/luke97ontologybased.html

[MB01] Generic schema matching with CUPID. Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. In Proceedings of the 27th International Conferences on Very Large Databases, pages 49--58, 2001. http://citeseer.ist.psu.edu/madhavan01generic.html

[MF00] The Chimaera Ontology Environment , 2000; D. McGuinness, R. Fikes, J. Rice, and S. Wilder.

[MK96] OBSERVER: An approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies, E. Mena, V. Kashyap, A. Sheth and A. Illarramendi, Proceedings of the 1 st IFCIS International Conference on Cooperative Information Systems (CoopIS '96), Brussels, Belgium, June 1996. http://citeseer.ist.psu.edu/article/mena00observer.html

[ML00] The Chimaera Ontology Environment. McGuinness, Deborah L., Richard Fikes, James Rice, and Steve Wilder. " Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI 2000). Austin, Texas. July 30 - August 3, 2000.

[MM02] Alexander Maedche, Boris Motik, Nuno Silva, and Raphael Volz. Mafra - a mapping framework for distributed ontologies. In Proceedings of the EKAW 2002, 2002.

[MW00] A Graph-Oriented Model for Articulation of Ontology Interdependencies. P. Mitra, G. Wiederhold , and M. Kersten. In: Proceedings Conference on Extending Database Technology 2000 (EDBT'2000), Konstanz, Germany, 2000. http://citeseer.ist.psu.edu/mitra00graphoriented.html

[NF99] An Overview of Active Information Gathering in InfoSleuth ;Marian Nodine, Jerry Fowler and Brad Perry. In Proceedings of the International Symposium on Cooperative Database Systems for Advanced Applications, 1999.

[NK03] Ontology Evolution: Not the Same as Schema Evolution, 2003; Natalya F. Noy, Michel Klein

[NM00] N. F. Noy & M. A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment In the Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000), Austin, TX. Available as SMI technical report SMI-2000-0831 (2000)

[NM02] N. F. Noy & M. A. Musen PromptDiff: A Fixed-Point Algorithm for Comparing Ontology Versions. In the Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002), Edmonton, Alberta, August 2002. Available as SMI technical report SMI-2002-0927 (2002)

[NM03] The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping, 2003; Noy N. F., M.A. Musen

[O03] Ontologies for Semantically Interoperable Systems, 2003; Leo Obrst

[OW03] Ontoweb deliverable 1.4: A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies, 2003

[PH99] The KRAFT Architecture for Knowledge Fusion and Transformation. A. D. Preece, K.-Y. Hui, W. A. Gray, P. Marti, T. J. M. Bench-Capon, D. M. Jones, and Z. Cui. In 19th SGES Int. Conf. on Knowledge-based Systems and Applied Articial Intelligence (ES'99). SpringerVerlag, 1999. http://citeseer.ist.psu.edu/article/preece99kraft.html

[PS00] Database integration: The key to data interoperability, 2000; C. Parent, S. Spaccapietra

[RB01] A Survey of Approaches to Automatic Schema Matching, 2001; E. Rahm, P.A. Bernstein

[RB01] A survey of approaches to automatic schema matching, 2001; E. Rahm; P.A. Bernstein

[SA03] Relationships at the Heart of Semantic Web: Modeling, Discovering, and Exploiting Complex Semantic Relationships , June 2003; Amit Sheth; I. Budak Arpinar; Vipul Kashyap

[SK03] Integrity and change in modular ontologies, 2003; Stuckenschmidt, H. and Klein, M

[SM01] FCA-Merge: Bottom-up merging of ontologies. G. Stumme and A. Madche. In 7th Intl. Conf. on Artificial Intelligence (IJCAI '01), pages 225--230, Seattle, WA, 2001. http://citeseer.ist.psu.edu/stumme01fcamerge.html

[SM03] Ontology Merging for Federated Ontologies on the Semantic Web, 2003; G.Stumme, A.Maedche

[TP03] Using Semantic Web Technology to Enhance Current Business-to-Business Integration Approaches, 2003; David Trastour, Chris Preist, Derek Coleman

[WV01] Ontology-Based Integration of Information - A Survey of Existing Approaches, 2001; H. Wache T. Vogele U.Visser et alii

[YP03] Using transformations to improve semantic matching, 2003; P.Z. Yeh, B. Porter, K. Barker